Towards Exascale Computing for Cosmological Simulations

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

James Douglas Potter

aus

Kanada

Promotionskomitee

Prof. Dr. Ben Moore	(Vorsitz)
Dr. Joachim Stadel	(Leitung der Dissertation)
Prof. Dr. Romain Teyssier	

Zürich, 2017

Contents

Chapter 1

Introduction

The fields of computational astrophysics has exploded over the last few decades, due mainly to the incredible (and seemingly unending) increase in the performance and capabilities of computers. The early 1960's saw machines capable of performing thousands of floating point operations each second. Today we have achieved a more than ten orders of magnitude increase in that performance as shown in figure 1.1.

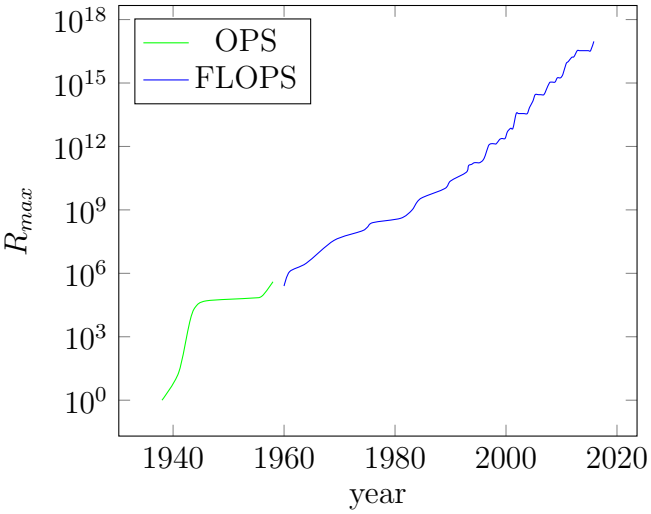


Figure 1.1: Supercomputing performance

This dramatic increase in performance has come with a corresponding lag in software development; Methods that worked perfectly well with the computing power available in 1990 now perform quite poorly in many cases. To understand why, it is useful to review the simulation method (in general terms) for computational science. The general process is as follows.

1. A set of "Initial Conditions" are generated. This could be a set of planets orbiting a star, a distribution of "dark matter", or a group of primates for an evolutionary biology simulation. There is a **time** associated with the initial conditions; for cosmological simulations it is the time since the big bang. We start early enough so that the system has no non-linear effects.

2. These initial conditions are evolved forward in time to some end state following a set of rules. For cosmological simulations we often evolve the system forward to the present time (i.e., today), but not always.

3. The resulting state of the system (and often many of the intermediate states) are analysed to identify physical quantities that are interesting, and that can provide us with insights into the system.

This technique is used where the system is too complex to analyse analytically, such as when the system exhibits non-linear or chaotic behaviours. Such simulations are normally taken to be statistically valid when measured quantities are said to have converged. In some cases a single simulation may provide enough information while in other cases an ensemble of simulations may need to be run.

When the field was just beginning, all three phases could be, and were performed on a single computer. As the size of the problems increased, the cost of the simulation started to dominate. A myriad of techniques were developed in order to accelerate this phase of the process, both improvements in the numerical techniques and the use of "parallel" programming techniques. It was possible to use hundreds, then thousands, and today even millions of computing elements to solve the problem with a corresponding speed-up[1].

Eventually, the very largest simulations started to produce results that were becoming increasing difficult, and in some cases impossible to analyse. Many of the analysis codes assumed that it was possible to simply read in the data, perform the analysis and output a reduced set of quantities. At some point it became impossible to read in the entire simulation output on a single machine; it was just too large. Additional techniques were developed where the analysis could be done as the output was read, but this does not work for all types of analysis.

Another problem was the amount of time it takes to perform the analysis. The amount of time it takes to analyse the outputs has grown to the point where it is now significant. For this reason, the analysis software has also started to use the full parallalism provided by modern supercomputers.

A final problem that has only recently started to plague large simulations is the data storage requirements. A single simulation can now require tens or hundreds of terrabytes (TB) of storage for each output, and simulations may need hundreds or thousands of outputs to achieve the necessary time resolution for the analysis. With so much data, a significant amount of time is spend writing and reading to storage. In many cases it is simply impossible or impracticable to store that much data as that amount of storage may not be physically available.

To achieve the necessary time resolution, while operating within the storage constraints, it is now necessary to perform analysis while the system is running, and output only the reduced set of quantities required. This requires that the set of required quantities be identified in advance as it is often not possible to reconstruct them without rerunning the simulation. This is an ongoing problem as it is quite difficult to build in *efficient* analysis into an existing simulation code.

One may think that the problem of generating initial conditions would remain free from these problems, but that is not the case. As the size of the simulations has grown, so has the size of the initial conditions. At some point it was no longer possible to fit an initial condition into the memory of a single node and parallel techniques were required.

[1] Using 100 compute elements can, in theory, decrease the time to solution by a factor of 100. In practice the speed-up is something less than a factor of 100.

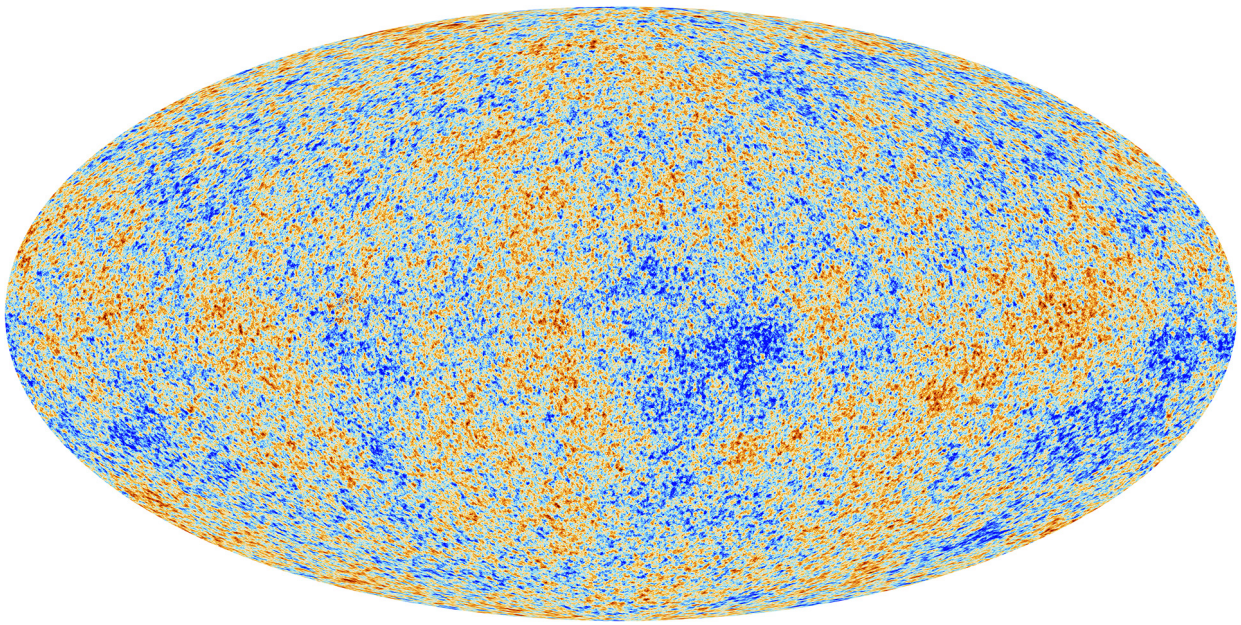


Figure 1.3: Cosmic Microwave Background map (courtesy ESA and the Planck Collaboration).

1.1 Cosmology

The focus of this work is large scale cosmological simulations, so it makes sense to review the motivation. Everything we "see" in the universe is made of of Baryons. It was then quite a surprise to discover that this accounts for less than 5% of the content of the universe.

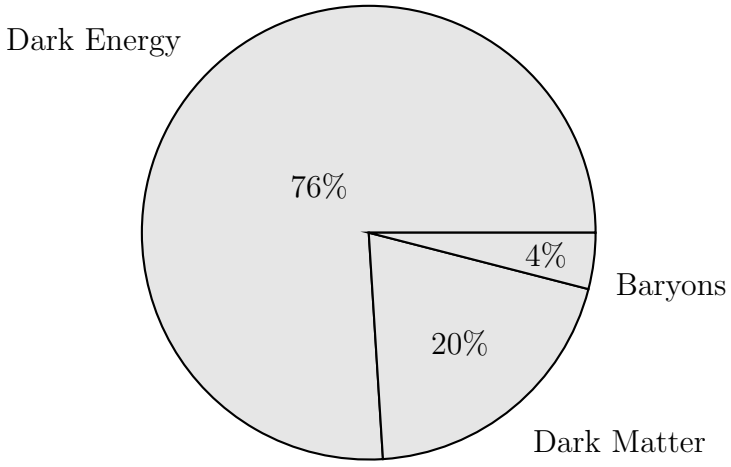


Figure 1.2: Content of the Universe

In 1933, the existence of dark matter was first proposed by Zwicky[220] to explain the orbital velocities of galaxies in clusters. The dominate theory today is the so called ΛCDM model because it is the most simple theory that explains the observables; most notably the Cosmic Microwave Background (CMB) and the accelerating expansion of the Universe.

In the very early Universe, the matter distribution was very nearly uniform. The density fluctuations have been very accurately mapped, first by Cobe, then by the Wilkinson Microwave Anisotropy Probe (WMAP), and subsequently in greater detail by Planck shown in figure 1.3. It should be noted that the fluctuations shown are on the order of 1 part in 100,000 and it is only after having undergone gravitation collapse that we find the dense structures that we see today.

Complementing the data we have collected from the CMB, the Sloan Digital Sky survey also provides information on scales up to $k \lesssim 0.2$.

On large scales ($k \lesssim 0.3$), we are still in the linear regime and it is possible to directly compare observations with theory and the agreement is excellent. In order to improve the precision of the known cosmological parameters, and to test the existing model, and potentially new models, we need additional information about even smaller scales. The difficulty is that a number of non-linear effect (such as gravity) collude to make it impossible to compare observations directly with theory as the effects cannot be calculated analytically. The measured power spectrum differs from the linear prediction because of these non-linear effects and the only way to compare theory with non-linear observations is through simulations.

To try to understand better the nature of dark matter and dark energy, the European Space Agency (ESA) has funded the Euclid Mission; a probe that will operate in L2 and which has been designed to very accurately measure the redshift of galaxies out to $z \approx 2$ (or about 10 billion years) which is the regime where dark energy has accelerated the expansion. The survey will focus on a section of sky that is free from light contamination for our solar system and the Milky Way galaxy. This area will encompass roughly 15,000 square degrees, or a little less than half of the sky. When complete, the survey will have produced tens of Petabytes of data consisting of hundreds of thousands of images with around 10 billion objects.

The smallest objects observed will have a mass of around $10^{10}h^{-1}M_{\odot}$. For simulations to resolve objects of this mass, the individual particle masses must be an order of magnitude lower (or $10^{9}h^{-1}M_{\odot}$). In addition, the simulation volume must be suitable large to avoid variations due to simulation cosmic variance which means that the simulation volume must be of order $4h^{-3}Gpc^{3}$. To achieve a particle mass of $10^{9}h^{-1}M_{\odot}$ in this volume, there must be 16384^{3} (or 4 trillion) particles. This work shows that it is currently possible to run such a simulation; in fact a 2 trillion particle simulation has already been run, and access to a supercomputer with sufficient memory is the only barrier. Supercomputers with sufficient memory already exist, and are becoming more prevalent.

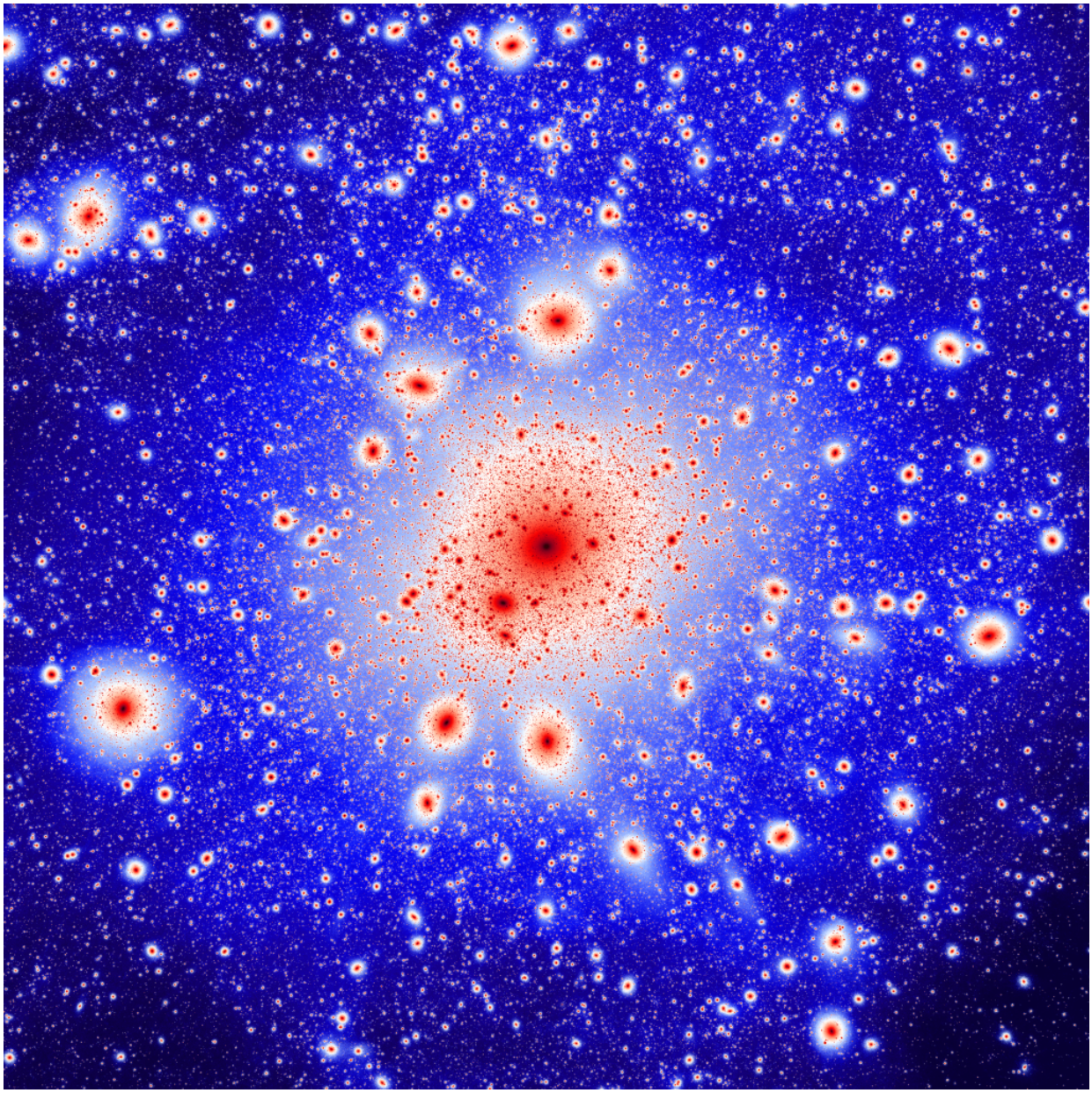


Figure 1.4: Milky Way sized dark matter halo. Colours indicate dark matter density and the hierarchical substructure is apparent.

1.2 Simulating Dark Matter

Much of the large scale structure of the Universe can be directly observed through, for example, the Cosmic Microwave Background (CMB). For large scales, the evolution has been linear so theory and observations agree very well. For smaller scales — below approximately 1 Mpc — linear theory breaks down and it is necessary to switch to modelling the system. The most powerful method available is the N-body approach. The Newtonian equations of motion for N particles is:

$$\boldsymbol{a} = \frac{d\boldsymbol{v}_i}{dt} = \sum_{j \neq i}^{N} G m_j \frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^3} \tag{1.1}$$

where $\boldsymbol{r}_i$ refers to the position of the i'th particle, m_j is the mass of the j'th particle, and G is the gravitational constant.

Dark matter is believed to be non-interacting, except through gravity, so it is useful to treat this as a collisionless fluid using the Collisionless Boltzmann Equation (also called the Vlasov Equation).

$$\nabla^2 \phi = 4\pi G \int d^3\boldsymbol{v} f(\boldsymbol{r}, \boldsymbol{v}, t) \tag{1.2}$$

where f is the mass in the volume element (or phase-space density).

The very first N-body simulation didn't use computers, instead in 1941 Holmberg[94] used a series of moving light bulbs as a proxy for gravity as both obey the inverse square law. Much later in 1970, Peebles[148] used 300 particles to simulate the Coma cluster with a direct $\mathcal{O}(N^2)$ technique. While N has increased dramatically for cosmological simulations making direct summation infeasible — one may argue impossible — the underlying technique is still used for a class of problems where integration accuracy is extremely important[76].

In the mid to late 1980's a pair of computational algorithms were discovered that reduced the complexity of gravity solvers from the intractable $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log_2 N)$. The first technique, the tree-code, was proposed by Josh Barnes and Piet Hut in 1986[14]. With this method, particles are arranged spatially using a tree data structure. Nodes in the tree contained a monopole approximation of the contained particles. If cells are sufficiently distant, then the cell can be treated as a single interaction. The hierarchy of the tree scales as $\log_2 N$ (for a binary tree), and thus leads to the $\mathcal{O}(N \log_2 N)$ scaling. Another $\mathcal{O}(N \log_2 N)$ technique was proposed in 1988[92] called Particle-Mesh (PM) where the mass of particles are assigned to cells of a mesh, and then the Poisson equation is solved on this mesh using the Fast Fourier Transform (which also has $\mathcal{O}(N \log_2 N)$ complexity). The PM methods suffers from the resolution being limited to the size of the underlying mesh whereas the tree code is self adapting by its nature. To overcome this limitation, a hybrid version combining direct Particle-Particle (PP) interactions for short range interaction with a PM technique for long range forces leading to the PPPM or P^3M[37] technique.

Although the $\mathcal{O}(N \log_2 N)$ behaviour of these codes was instrumental in the explosion of N-body simulation codes, it is possible to do even better. Greengard and Rokhlin[75] proposed a technique in 1987 now called the "Fast Multipole Method" (or FMM) which has algorithmic complexity of $\mathcal{O}(N)$. Due perhaps to the difficulty of implementing the technique efficiently, and also because of the entirely satisfactory performance of tree codes, it was more than two decades until a cosmological N-body code implemented it[196]. As

the "N" in N-body has crept up into the trillions, the $\mathcal{O}(\log_2 N)$ factor of tree codes has continued to grow and have had an increased effect on the run-time. The most recent, and largest to-date simulation[151] of 2 trillion particles was run with PKDGRAV3 which is the only cosmological code to use FMM. Naïvely, when N is 2 trillion, an $\mathcal{O}(N)$ code should be 41 times faster than an $\mathcal{O}(N \log_2 N)$ code as $\log_2(2 \times 10^{12})$ is 41, but there are other factors involved. An FMM code does not come for free; although it scales as $\mathcal{O}(N)$, each operation involves additional calculations when compared to a simple $\mathcal{O}(N \log_2 N)$ tree code. The FMM code starts to win when N reaches millions of particles.

1.2.1 Analysis

Just as important as the actual simulation is the analysis of the output as this is where the insights into cosmology are revealed. Traditionally, because of the float-dense nature of the gravity solvers, they tended to dominate the run time; it took days or weeks to run a simulation whereas, the analysis could take only hours. Today the situation is quite different for these large cosmological simulations.

Modern computers are exceedingly good at performing calculations, and the performance has been increasing exponentially (see figure 1.1). Storage on the other hand has seen only a modest increase in performance over the same period of time, and although the performance continues to increase, it does not do so exponentially and thus storage performance is quickly being outpaced by computation. It is now possible to generate data faster than it can be saved, and the situation will only get worse. The typical work-flow for simulations has historically been to first generate all of the data for the simulation and save it to disk. Only after the simulation has complete is the bulk of the analysis performed. Some analysis requires only a single snapshot (perhaps at redshift zero), while other analysis, such as the generation of merger trees, may require between 100 and 400 outputs. For the four trillion particle Euclid simulation, each of these outputs is **at least** 100 Terrabytes, and would need between 10 and 40 Petabytes of storage. The Piz Dora scratch file system, a state of the art Lustre parallel file system, has a peak performance of 50 Gigabytes per second. If it was large enough, which it is not, then it would take between two and eight days just to save the data.

For this reason, future simulations will perform the bulk of the analysis "on-the-fly", writing only the reduced data. While this has been done to varying degrees in the past, it will soon be cheaper the simply rerun the simulation if a different set on analysis needs to be performed. It is useful to think of the analysis "modules" as "instruments", and the simulation as an "experiment" in this context.

Some instruments are quite simple; the measurement of the power spectrum (related to the two-point correlation function) is relatively straightforward to include. The particles masses are projected onto a grid using a mass assignment algorithm (such as NGP, CIC or TSC) and an FFT is performed to recover the power spectrum. This process can be repeated at fixed intervals throughout the simulation to track how the power spectrum changes with redshift. The generation of a light-cone is another conceptually straightforward analysis task — particles are identified as being included in the light cone and project onto a spherical grid (typically using a Healpix[72] mapping), and output in shells. Halo finding is a more complicated process, but as shown in chapter 5 it can be effectively parallized and integrated into a simulation code.

Other analysis tasks will need to be integrated into the simulation framework, and this will continue to be a challenge as the code must be robust, efficient and effectively parallized lest the simulation be negatively effected.

Chapter 2

Computational Challenges

Numerical simulations have, from the very beginning, pushed the limits of what is computationally feasible. With pure N-body, the resolution directly affects how accurate the simulation is, and how well structure can be resolved. For this reason, increasing the number of particles is almost always the easiest way to improve the quality of the result. The problem is, of course, that it takes longer and longer to run the simulation.

There are a number of computational techniques that can be used to decrease the overall runtime making it possible to finish a simulation in a reasonable amount of time. These will be discussed in the following sections.

2.1 Computational Complexity

It is useful at this point to review the concept of computational complexity. Some of the largest gains in run-time have involved algorithmic changes to decrease the computational complexity. The computational complexity of an algorithm can be described using "big-O" notation. As an example, if you have a list of numbers and you want to see if it contains a certain value, the obvious way is to look at every element. If you are lucky it will be the first one, and if you are unlucky it will be the last, so it make take only 1 try, or it make take N. Normally one looks at the average and worst cases. If you do multiple searches, you will eventually have a distribution of search times which will cluster around the average.

In our example of the search, you need to look at half of the values on average or $\frac{1}{2}N$ while the worst case is N. We are usually only concerned with how the algorithm scales when you increase N, so with big-O notation, the prefactors are dropped. In this example, the average search time is $\mathcal{O}(N)$ and the worst case is also $\mathcal{O}(N)$.

Let's now consider how one could sort a list. A naive approach is to use what is called a selection sort. We scan through the list to find the smallest number and move it to the front. We then scan the remaining elements for the next smaller and so on. We need to look at $N + N - 1 + N - 2 \ldots \approx N^2$ elements before the sort is done. This means the selection sort is $\mathcal{O}(N^2)$. The most commonly used sorting algorithm is the quick sort which is $\mathcal{O}(N \log_2 N)$. It is even possible to sort in $\mathcal{O}(N)$ time if you know something about the distribution of keys in your input. As a final example, if we know that the list has

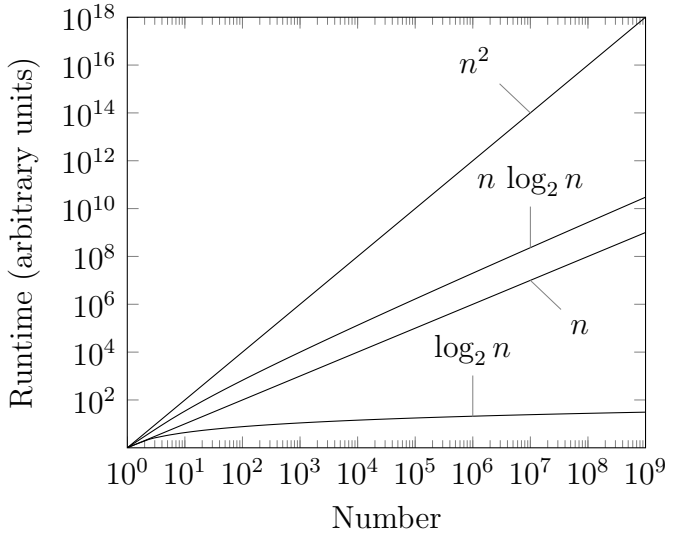


Figure 2.1: Scaling for different algorithmic complexities

already sorted, it is possible to search for elements using a "binary search". This reduces the complexity of the search to $\mathcal{O}(\log_2 N)$.

So why does this matter? Figure 2.1 show the scaling for different complexities. If an operation were to take 1 nano-second, then the runtime is as follows for the various computational complexities for the given number of "items", be it elements of a list, or particles in a simulation.

	1 thousand	1 million	1 billion	1 trillion
$\mathcal{O}(\log_2 N)$	10 nanoseconds	20 nanoseconds	30 nanoseconds	40 nanoseconds
$\mathcal{O}(N)$	1 microseconds	1 millisecond	1 second	17 minutes
$\mathcal{O}(N \log_2 N)$	10 microseconds	20 milliseconds	30 seconds	11 hours
$\mathcal{O}(N^2)$	1 millisecond	17 minutes	32 **years**	32 **mega-years**

As can be seen in this table, the differences are small when "N" is small, but things change dramatically as N grows. In particular, any algorithm that is $\mathcal{O}(N^2)$ becomes disastrous when N gets too large; often this is before even 1 million. Examples of $\mathcal{O}(N^2)$ algorithms abound. Initial condition generation can be done by convolving Gaussian random noise with a transfer function. Convolution (in real space) is an $\mathcal{O}(N^2)$ operation. Gravity calculations formally require all pair-wise interaction to be evaluation which is again an $\mathcal{O}(N^2)$ operation. Fortunately for us, alternate algorithms have been developed to reduce the complexity making large simulations feasible.

2.1.1 Space Complexity

It is also worth noting that algorithms have a space complexity, or how much memory they require. Normally, it is necessary to have all items in memory so the space requirements are at least $\mathcal{O}(N)$. There exist algorithms where you can trade of memory for computation so by using $\mathcal{O}(N \log_2 N)$ memory for example, an algorithm could go from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. In other cases, poor choices result in algorithms using more than $\mathcal{O}(N)$ memory when they

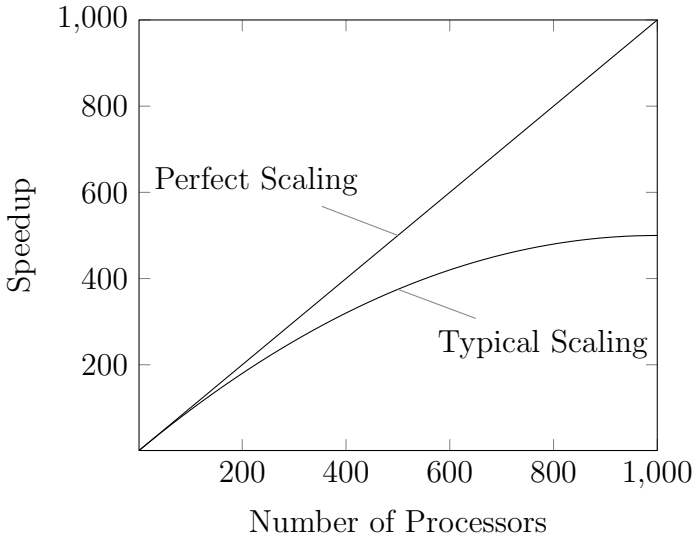


Figure 2.2: Scaling

could easily be replaced with a better one. During development one needs to keep in mind the order of the algorithm both in terms of computation, but also in terms of memory usage.

2.2 Parallelism

Once we have an efficient algorithm, we may want to make it faster. A certain computer simulation is running on a workstation, but it's going to take one year to complete. If we had 100 computers, and we could use them all together it should be 100 times faster. Our simulation that was going to take a year would now take less than four days. The idea is a simple one, but implementing this turns out to be quite difficult.

2.2.1 Scaling

To measure how well a code has been parallelized, we look at the scaling. With "perfect" scaling, the speedup scales linearly with the number of processors used. In practice, this is difficult or even possible to achieve. As seen in figure 2.2, the typical scaling starts the same as the perfect scaling but eventually drops off. It can even turn over, requiring even more time when adding additional processors.

The shape of the scaling curve will change from algorithm to algorithm, and also based on the input to the algorithm. For example, highly clustered simulations scale much worse than large uniform boxes. More on this in Chapter 4.

Strong Scaling

Strong scaling is the hardest type of scaling to achieve. With strong scaling, the problem size remains fixed while the number of processors grows. As an example, a simulation of a

galaxy is run on 100 processors, but it is going to take two months. With perfect strong scaling it would be possible to run the same simulation on 1000 processors and reduce the run-time to six days.

Weak Scaling

Weak scaling is easier to achieve, but by no means easy. With weak scaling, the size of the problem is scaled with the number of processors. In this scheme, a 500 Mpc box with 1024^3 particles is run in 1 week using 10 processors. Now we run a 1000 Mpc box with 2048^3 particles using 80 processors. We would expect it to also take a week to run if we had perfect weak scaling.

2.2.2 Domain Decomposition

The first task in any parallel code is the domain decomposition. We want each processor to do its share of the work, so we need to divide the work into processor sized bits. The mechanism chosen to divide the work strongly depends on how the algorithm will operate on the data.

Load imbalance

The primary requirement of domain decomposition is that the work needs to be balanced. Ideally all processors will take the same amount of time and finish at the same time, but this can be hard to achieve except for extremely simple algorithms. Take for example an N-body simulation. Does one divide the work equally by number of particles? To first-order this is a reasonable approach, but particles in high density regions have more "work" than those in low density region. The amount of work changes over time, so no approach will achieve perfect load balance.

Locality

Another problem with certain algorithms, including N-body is that each particle requires information from other particles. With N-body, more data is required from "near-by" particles but some information is still required from far-away particles. A common strategy in this case is to divide the particles spatially so that nearby particles are on the same processor. Access to nearby particles is immediate, while distant particles require communication. Too much communication and the run-time will dramatically increase (communication is much slower than computation).

A technique to overcome this communication latency is to make a copy of remote data locally. In the extreme case, all particles could be on all processors, but this is feasible only for very small numbers. A more common approach is to copy only the needed parts; some algorithms need only adjacent regions so "ghost" cells or particles are created. In other cases, it may be possible to determine, in advance, what will be required and request it in a single communication phase. This eliminates much of the communication latency, but with sufficiently complex algorithms it becomes difficult or impossible.

Multi-Stepping

Another common technique is multi-stepping. To integrate an orbit accurately, one needs to take approximately 30 steps per orbit. Different orbits take different lengths of time; Mercury orbits in 88 days while Neptune takes 165 years. This means that Mercury needs to be stepped 684 times before Neptune needs to be stepped once. The same idea applies for large N-body simulations. At any given time, only some of the particles are "active". The inactive particles are still needed as part of the calculation of the gravitational potential.

We would like to evenly divide the "active" particles amongst all of the processors, but if we stick to a spatial distribution, it soon becomes impossible to divide both the works, and the number of particles. If we divide active and inactive particles separately then locality suffers.

2.3 Hardware Acceleration

For several decades, the speed of processors followed Moore's law simply by increasing the clock speed and number of transistors. At this point, that trend is coming towards an end, but performance gains continue by on-chip parallelization. For the main processors, this takes the form of "short vectors"; SSE or AVX on Intel processors and Altivec on IBM. Supercomputers are also increasingly coming equipped with dedicated accelerators such as nVidia GPU cards, or the Intel Xeon Phi. Effectively using this hardware can result in up to an order of magnitude increase in performance.

2.3.1 Single Instruction, Multiple Data (SIMD)

The mainstream Intel processors all have a series of SIMD extensions that allow multiple numerical calculation to take place simultaneously such that it is possible to do 2, 4 or 8 floating point operations in the same time as one. The limitation is that the operation has to be identical; only the data can be different. Compilers have become increasingly good at automatically converting single operations into vector operations (a process known as vectorization), but it is imperative that the data be organized in a way that makes this possible and efficient.

2.3.2 Single Instruct, Multiple Threads (SIMT)

The nVidia line of graphic processors includes a SIMT engine that can be used to calculate. Some of the fastest Supercomputers in the world have a dedicated GPU attached to each node. Using the device effectively can be tricky, but can increase the performance of applications by up to an order of magnitude, though a 2 or 3 fold increase is more usual.

2.3.3 Single versus Double Precision

Numerical simulations very often use double precision floating point numbers for their calculations because of numerical artefacts or precision problems that occur with single

precision. For some time, double precision calculations were slower than single, but today it makes no difference. No difference, except when talking about SIMD or SIMT acceleration. With SIMD, the double and single precision instructions take the same time, but with single precision you can do twice as many of them! In the case of SIMT (nVidia Tesla for example), there is separate hardware, and the double precision performance can be between three and thirty-two times worse depending on which model is used.

For this reason, using single precision calculations will increase the performance. Individual algorithms must be carefully analysed and tested to verify that single precision will not cause numerical problems.

2.4 MPI and Threads

When parallalization across nodes in desired, the most ubiquitous method is to use MPI (Message Passing Interface). The MPI specification is mature, well testing and supported on all systems. If the goal is to parallalize on-node, then a common technique is to use threads, either using OpenMP directives, or by using an explicit threading model such as pthreads.

With threads, it is possible to schedule work which helps with load balancing, and all data is "local" so data copying can be minimized or eliminated. Obviously, threads restricts the problem to a single node with a relatively small amount of cores.

There exists a third, hybrid method. One starts a single MPI process for each node, and the MPI process uses all of the cores via some threading technique. This is the superior method, but it is much more complicated. One has to write both the MPI version of the algorithm, as well as the threads version, plus one has to tie the two together. Sometimes this is simple, but more often it is quite complex.

The advantage of the hybrid approach is that it can offer an immediate increase in performance, and has other benefits discussed below.

2.5 Large Numbers of Processors

With modest numbers of processors — hundreds or even a few thousand — most parallel codes perform quite well and scale reasonably. Once the number of processor reaches tens or hundreds of thousands then a new problem arises. There are usually things in the code that scale either directly, or indirectly with the simulation size[1], or with the total number of processors or tasks used. This will eventually result in too much memory being used and placing an upper limit on the size of the simulation.

2.5.1 MPI Overhead

The number of MPI processes tends to scale with the simulations size. If you need 1000 processors for a simulation of a certain size you will need 10,000 for a simulation that

[1] e.g., number of particles

is ten times larger. MPI codes commonly need to allocate communication buffers, and sometimes there needs to be one or more buffer for each remote node. These buffers can add up quickly with large numbers of remote processes, so care must be taken to avoid this practice if possible. Additionally, the MPI implementation itself may allocate buffers in this way internally.

By using a hybrid MPI/threads code the buffer requirements can be reduced by a factor of ten or twenty, but this only pushed the problem out to larger numbers of processors.

2.5.2 Master Nodes

Another common issue is with master nodes. It is tempting to collect summary information on a single node, or distribute common shared data to all of the nodes. This isn't ordinarily a problem unless this information again scales with the number of nodes. A group finder could keep a list of all groups it finds on a single node for example. The number of groups is much smaller than the total number of particles, but does scale as the total number so eventually there won't be enough memory left on that node. It's much harder to develop an algorithm where no single node has complete knowledge.

Chapter 3

Initial Conditions

One of the main challenges with cosmology is trying to connect the physical parameters that describe our universe with observations. The process is relatively simple on large scales, as the dynamical processes are still linear. On smaller scales, the various physical processes become more and more non-linear, and the only way to compare with observations is through simulation. The process begins with the generation of a realization of the Universe early enough that no non-linear effects have occurred. This strongly depends on the smallest scales being simulated; for a cosmological simulation a starting redshift of 50 may be adequate while for a zoom-in simulation of a galaxy we may need to start at redshift 100 or 150.

A few codes have been developed over the years, but probably the most successful was the GRAFIC[22, 23] series of codes. These codes were written in FORTRAN, and ran only in serial. For large simulations, including GHALO[196] (see chapter 7) it was not possible to generate the necessary initial conditions as the authors did not have access to a single machine with sufficient memory. Running in serial would also mean that even with sufficient memory the process would take days. Other more ambitious projects including the Euclid Flagship Simulation[151] (see chapter 6) would be impossible as the generation of the initial conditions require approximately 8×10^{13} bytes (80 PB) of memory. The exists no single server with amount of memory and the process would have been too time consuming (of the order of two months).

The GRAFIC1 code – a code to generate cosmological boxes of fixed resolution – was later enhance in mpgrafic[157] to use thread based parallization. This sped up the computation speed by a factor of 20 or more, but does not address the memory limitation. The mpgrafic code also did not provide a GRAFIC2 replacement which makes it unsuitable for zoom simulation such as GHALO.

The earlier methods such as GRAFIC used a first order scheme known as the Zel'dovich approximation[217] to generate particle positions given density fluctuations. It has been shown[171, 39] that using a second order scheme improves the accuracy of the initial conditions. The MUSIC[80] code overcame some of these restrictions by using second-order perturbation theory, and by providing support for zoom type simulations, but is still restricted to a single node with OpenMP parallelization.

In the following sections I describe the various challenges to generating initial conditions at scale, and how they were overcome.

3.1 GRAFIC++

The GRAFIC++ code was originally developed to generate the initial conditions (ICs) for the GHALO simulation. The existing tools were not sufficient, as the sheer size of the IC made it impossible to fit on a single node. Even today, GRAFIC++ with MPI support is capable of generating the very largest ICs for zoom simulations. Let us recall the basic process for generating ICs.

- Random variates (random numbers) are generated using some pseudo-random number generator.
- These variates are transformed to normal deviates (Gaussian white noise) using the Box-Muller[29] transformation.
- Convolve this signal with the transfer function to obtain a constrained realization.

For zoom simulations (where sections of the volume have improved resolution), the process becomes more complex.

- Generate new, higher resolution deviates for the sub-volume.
- Constrain them to match the main volume using the Hoffman-Ribak[93] method.
- Perform the convolution as before (but with non-periodic boundary conditions).

The point of non-periodic boundary conditions is important; cosmological ICs are normally periodic. A convolution of two signals is an $\mathcal{O}(N^2)$ operation which makes it computationally infeasible, however a convolution in real-space is a simple multiplication in k-space so a simple $\mathcal{O}(N \ln N)$ FFT can replace the convolution. The FFT is by nature a periodic function so handling non-periodic signals can be tricky. The approach taken by GRAFIC2 was to double the FFT in each dimension resulting in a factor of 8 more memory required.

3.1.1 General Considerations

The goal of GRAFIC++ was to generate ICs that could be run on the very largest supercomputers. Realistically this means that it must be capable of running on the specific supercomputer in question as that is the only way to guarantee it will scale. The most prevalent technology used today to parallize codes on supercomputers is MPI and GRAFIC++ uses it. It was also completely recoded in C++. The orginal GRAFIC code was also 32-bit, and some of the file formats reflect this. The new code must generate ICs with more elements than a 32-bit code can handle (2 or 4 billion) and thus special care was taken.

3.1.2 Random variates

The generation of truly random numbers is a topic that is both broad, and beyond the scope of this document. It is sufficient to say that when we say random we mean simply that when we generate a sequence of numbers, that they are uncorrelated. Ideally, we would like to be able to reproduce the sequence at will from a give starting seed.

The original GRAFIC programs used a subtract-with-borrow lagged Fibonacci generator[127] to generate random numbers. This generator works well, in the sense that it generates uncorrelated variates, but suffers from two limitations. First, the generator performs many calculations using floating point numbers. Any slight differences in intermediate results due to, for example, rounding differences caused by machine architecture or compiler changes reordering operations, will eventually result in the random number stream diverging from previous streams. The nature of the stream being uncorrelated exacerbates the problem — and deviation results in a completely different stream from that point on. This was overcome in the original version of GRAFIC by saving the random noise to a file for use in subsequent runs if necessary. This is not always practical due to the size of the files, and it not actually necessary. The second more serious limitation for us is that there is no way to generate random numbers in parallel; random numbers must be generated by a single processor and sent to others lest the sequence depend on the number of processors used – a process that can be quite time consuming.

This random number generator has been replaced with an implementation of the Mersenne twister [129] algorithm. This algorithm uses only integer math and bit manipulation which makes it much less susceptible to divergent behaviour. In principle, any machine that uses standard 32-bit two's complement integer arithmetic will generate the same sequence of random numbers for a given input seed.

In order to perform this operation in parallel, a compound key is used; the first part is the random seed provided by the user to indicate which random realization they desire. The second part of the key is the z-dimension of the grid. As the work will be divided into slabs along the z-dimension, this allows individual processors to generate their random fields independently, and the same set of variates will be generated independent of the number of processors used.

3.1.3 Random deviates

As previously mentioned, the random variates (uniformly distributed between 0 and 1) are transfored into random deviates (mean of zero and variance of 1) using the Box-Muller transformation. The original GRAFIC codes used the Cartesian form of the Box-Muller [29] transformation. Given two uniform random numbers, x_1 and x_2 two Gaussian random numbers y_1 and y_2 can be generated as follows.

$$y_1 = \sqrt{-2\ln x_1}\cos 2\pi x_2 \tag{3.1}$$

$$y_2 = \sqrt{-2\ln x_1}\sin 2\pi x_2 \tag{3.2}$$

This form of the transformation suffers from the fact that the calls to sin and cos are quite slow, and there are numerical stability problems when x_1 is very close to zero. Consequently, the new code uses the *polar* form[54] of this transformation. An x_1 and x_2 are chosen in

the closed interval $[-1, +1]$, but rejected if w is greater or equal to one, or is zero where w is defined to be:

$$w = x_1^2 + x_2^2 \tag{3.3}$$

The value w will be accepted approximately 78.5% of the time. The random numbers are then generated as follows.

$$w' = \sqrt{-2 \ln w / w} \tag{3.4}$$

$$y_1 = x_1 w' \tag{3.5}$$

$$y_2 = x_2 w' \tag{3.6}$$

3.1.4 MPI Parallization

One of the first decisions made when using MPI to distribute calculations is the best way to organize the data both on and off node. As the FFT algorithm is central to this technique, the data was organized to use the FFTW[67] layout as the other algorithms could be easily adapted. This results in the 3-D grid being decomposed into "slabs" along the z-dimension, with each process getting one or more slabs. The slab-decomposition minimizes data movement between nodes. There are other decomposition schemes such as the pencil decomposition or block decomposition that have other benefits, but that require cross-node communication for each transpose.

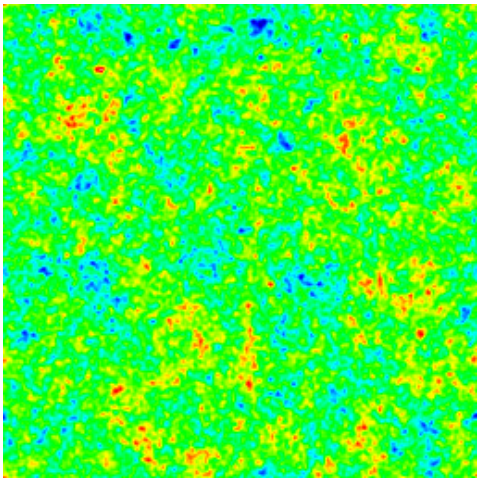
The only major disadvantage to the slab decomposition is that the number of nodes that can participate in the FFT is limited to the total number of slabs (or the size of the FFT in the z-dimension). This implies that a single slab must fit into the memory of a node; for a node with 32 GB of RAM (common today), this limits the size of the FFT to roughly $90,000^3$. A single slab of $90,000 \times 90,000$ would require 32 GB, and the entire FFT would need 90,000 **nodes**! The current limiting factor today is the number of nodes and total RAM available and not the slab decomposition scheme.

Each processor operates independently for the most part by, for example, generating the random numbers independently as previously discussed. There is, however, an important exception. The Hoffman-Ribek method requires the calculation of an average over the fine grid. The fine grid cells may be located on different processors, or even on different nodes. The order of additions is also important if a numerically identical result is to be obtained independent of the number of processors, so care is taken to maintain this ordering.

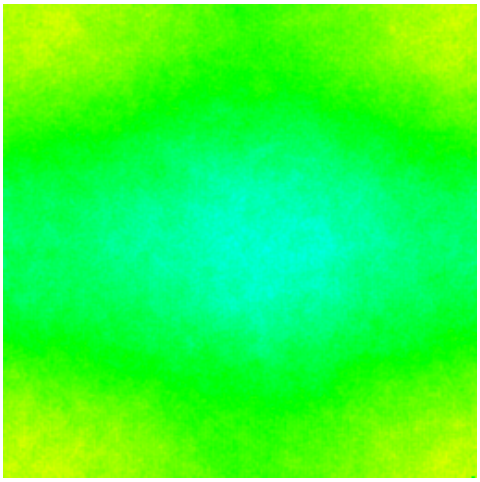
3.1.5 Accuracy

The first code of a given type always has a more difficult time as without anything to compare to, it is more difficult to show accuracy. As it is possible to feed the same random noise into both GRAFIC and GRAFIC++ it is easy to compare the results. Figures 3.1 and 3.2 show a slice of the density field for the periodic box and a refined region, along with the corresponding differences between GRAFIC and GRAFIC++. Note that the scale of the difference is 10^{-5} which is close to float round.off.

As the technique is based on the original GRAFIC code, it makes sense to compare the two. For a uniform cube, figure 3.1 shows the density of a slice of the volume as well as the absolute difference in the density between the two codes.

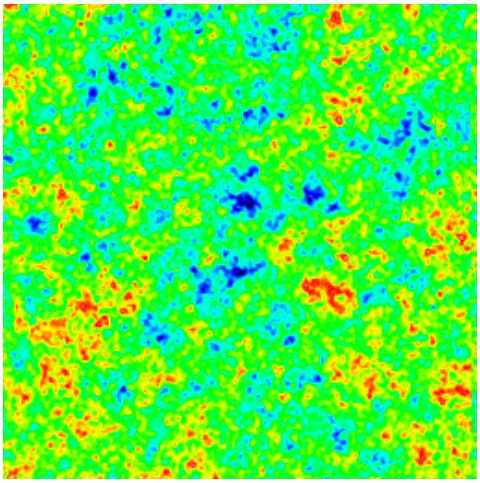


(a) Density slice. Scale of density fluctuations is between ±0.5.

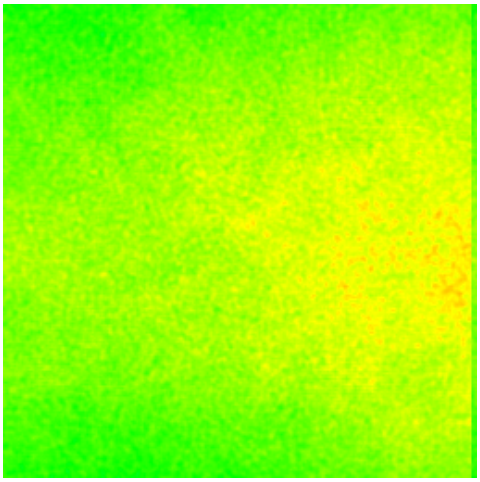


(b) Differences between GRAFIC and GRAFIC++ with absolute error between $\pm 10^{-5}$.

Figure 3.1: Density slice for the main box a 256^3 IC.



(a) Density slice. Scale of density fluctuations is between ±0.5.



(b) Differences between GRAFIC and GRAFIC++ with absolute error between $\pm 10^{-5}$.

Figure 3.2: Density slice for a refined region for a 256^3 IC with absolute differences between GRAFIC2 and GRAFIC++ shown.

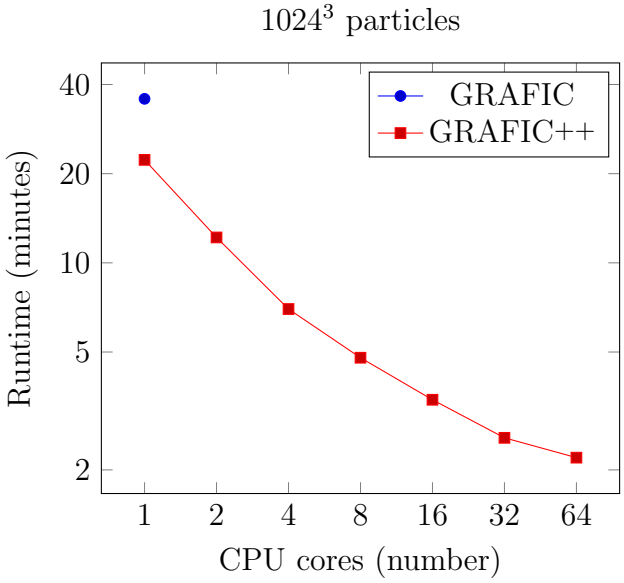


Figure 3.3: Scaling for GRAFIC++ compared to GRAFIC.

3.1.6 Performance

Figure 3.3 shows run-time for the original GRAFIC and GRAFIC++ for the same IC.

3.2 Refinement

We first generate a low-resolution, periodic grid of size suitable for the type of simulation. This volume is simulated to $Z = 0$ (or other suitable point of interest), and a halo of interest is identified. Typically this is done by running a group finder on the entire simulation volume and identifying, either automatically or guided, a suitable halo. For example, if the goal is to simulate a Milky Way like dark matter halo, the selected halo should have the correct mass (approximately $10^{12}\,\mathrm{M}_\odot$). It may be useful to provide other constraints such as requiring that the halo be "isolated" for example.

To increase the resolution, or "refine" this object, it is first necessary to locate it in the initial conditions. Naively, this can be done by taking every particle in the halo at $Z = 0$, and noting where it is in the initial condition. The smallest box that contains all of the particles would be the area of region. An example Lagrangian region is shown in figure 3.4.

Because halos are formed through gravitational collapse, this region of interest will be much larger in the initial conditions. As an example, the dark matter halo of the Milky Way is of order 350 kpc, while the extent of the area in the initial conditions is between 5 and 10 Mpc in each dimension.

To improve the quality of the high resolution simulation by avoiding mixing, and other effects caused when heavy (lower resolution) particles interact closely with lighter particles, the following step is employed to improve the quality of the initial Lagrangian region. The reason for this step will be become clear later. An important side effect of the simulation process is a time series of the particles, that is to say, we know the particle positions not only at $Z = 0$, and in the initial conditions, but also at many intermediate times. We

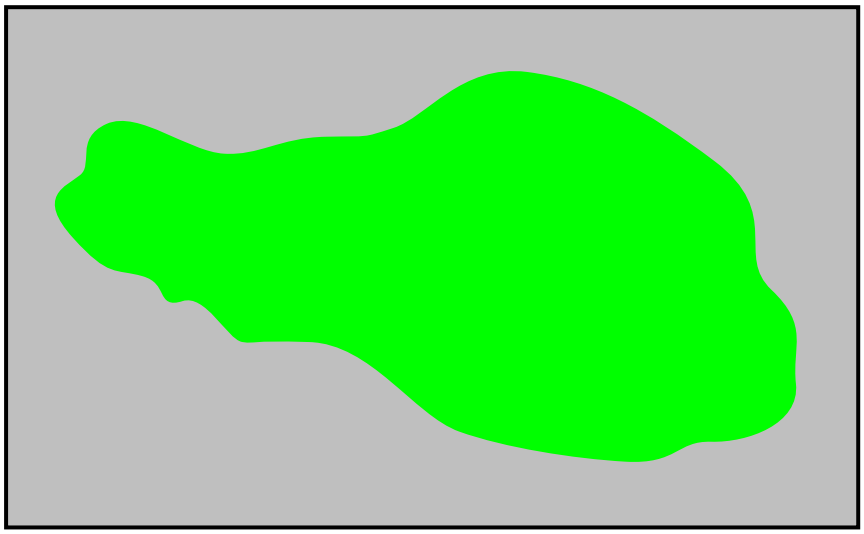


Figure 3.4: High resolution region with the Lagrangian region shown in green. The grey area included a buffer region to avoid mass-mixing.

analyse each of these time series (also called outputs), and identify any other particles that come in close proximity to the previously identified halo particles. Any such particles also contribute to the shape and extent of the final Lagrangian region. The choice of how close is a free parameter, but is typically a few times the mean inter-particle separation.

We take the Lagrangian region from the low resolution simulation shown in green in figure 3.6. Both the Lagrangian region (green) and a small buffer region around it (dark grey) remain at the very highest resolution. All particles outside this region are degraded by combining multiples of them together to form a single particle. This is done by finding the center of mass position, mass and velocity and is typically done for particles that would have been aligned to a courser grid (so $2 \times 2 \times 2$ or 8 particles for example).

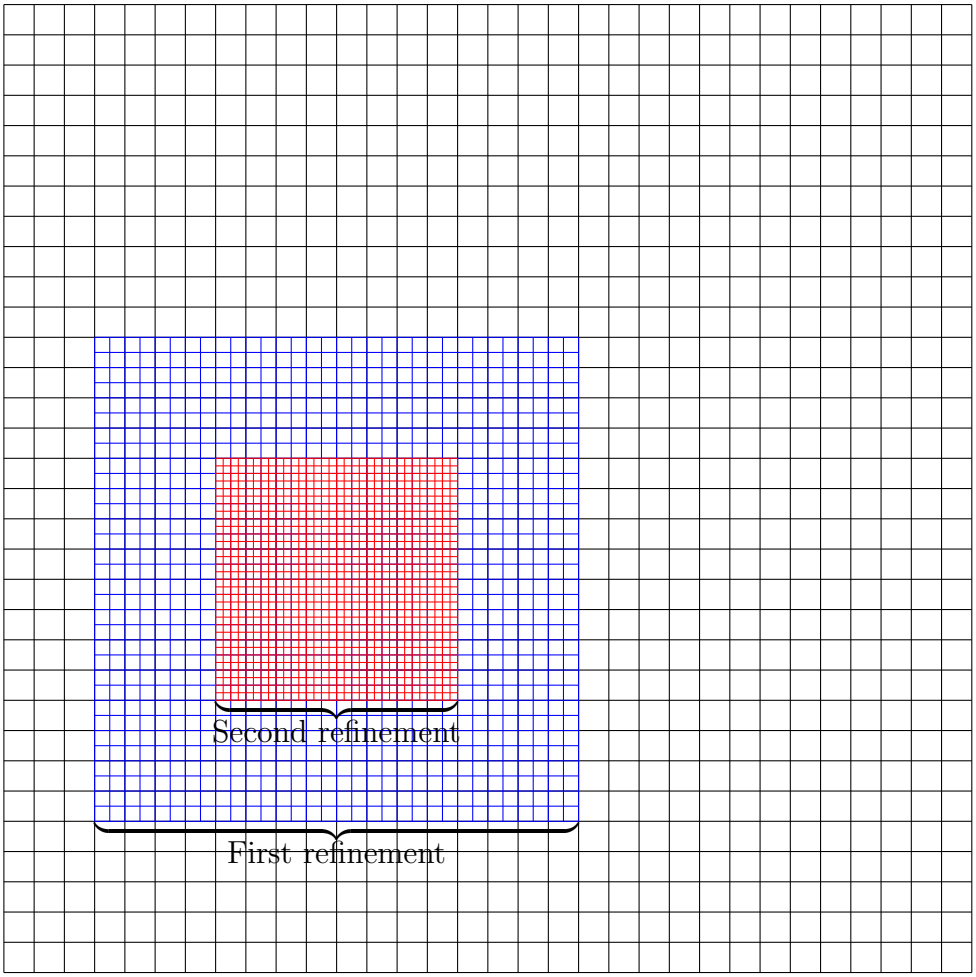
The buffer region is used to compensate for slight variations between simulations of different resolutions; the goal of which is to avoid particle mixing where "heavy" particles migrate into the high resolution region (with the "light" particles). Some level of mixing is unavoidable near the outside of the final bound structure but by choosing an appropriate buffer regions (typically a few course cells), this effect is minimized.

It should be also noted that this buffer region typically forms a shell around the bound structure and as such is part of a low density region. Low density regions are much less computationally challenging to simulate, so the cost of adding a high resolution buffer region is quite small compared to the overall runtime.

The particles that form this halo are traced back to the initial condition, and their Lagrangian region is identified as shown in figure 3.4.

The complete Lagrangian region is refined by adding small scale power. The simulation is rerun with the increased resolution. This process can be repeated, but the size and position of the final region must obey the limitations of the method. See figure 3.5.

Figure 3.5: Nested refinement regions



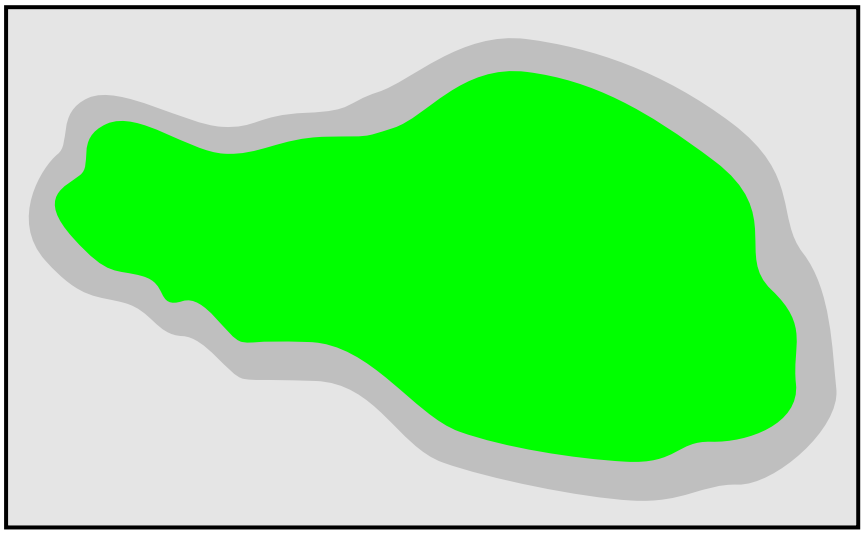


Figure 3.6: High resolution region with Lagrangian region highlighted in green. The dark gray shell remains in high resolution, while the light gray region is derefined to a lower resolution.

3.2.1 Further Derefinement

This technique results in a set of hierarchical cube-shaped nested regions with increasing resolution. It is perfectly valid to run a simulation with these regions, but it is wasteful to simulate regions outside the area of interest at a high resolution. In order to reduce the particle count by an order of magnitude, we combine particles by a process called derefining.

In figure 3.6 we see the original Lagrangian region in green as before, and the high-resolution cube in light gray. We identify a shell around the Lagrangian region by identifying grid points that are within a certain distance from any particle found in the Lagrangian region. The choice of distance is arbitrary, but usually around 10 course cells distance is adequate. This shell is indicated by the dark gray region in the figure. The dark gray region remains at the same resolution as the Lagrangian region, while the outer box is derefined by replacing multiple particles with a single higher mass particle — essentially a monopole. This is done by taking a $2 \times 2 \times 2$ or $3 \times 3 \times 3$ set of grid points and combining them.

This process can be repeated several times with more shells until the computation involved in the outer shells becomes insignificant. As the outer region exists only to provide a tidal field for the halo being simulated, the end result is unaffected.

3.3 PKDGRAV3

The requirements for initial condition generation for large cosmological simulations consisting of trillions of particles or more are somewhat different than for zoom-simulations.

- The size of the particle data can reach a size where reading and writing the data can take 30 minutes to an hour.

- The slab decomposition breaks down with a pure MPI based approach. The 2-trillion particle Euclid Flagship simulation required 4000 nodes for enough memory which was 32,000 cores. The 12600^3 simulation could only use (at most) 12600 slabs which is less than the 32,000 cores available.
- Second order is probably required for these types of simulations.

To address these shortcomings, a specialized version of the IC generator was built directly into PKDGRAV3. It was based on top of the high performance parallel engine of PKDGRAV3 and solved the problems in the following way. As PKDGRAV3 is a hybrid code using a single MPI process per node, and multiple threads within the node, the FFT was changed to this model. Each node now hosts one or more slabs, and within the node the FFT is accelerated by using multiple threads. This means that a single 64GB node could handle a slab of 40000^3, and with 40000 such nodes it would be possible to generate the initial condition for a 64-trillion particle simulation. This IC generator can also generate first- and second-order results. In addition, as the IC is generated by the code when first run, the time to save and reload (more than an hour) is avoided.

The entire Euclid flagship simulation took approximately 80 hours on 4000 nodes of Piz Daint and the generation of the second order initial conditions took only 155 seconds.

3.3.1 Second Order

The second order calculation involves many terms. Traditionally the calculation goes as follows.

- Calculate white noise field.
- Calculate three first order displacements.
- Calculate six second order terms.
- Combine second order terms into the source term.
- Calculate three second-order displacements from the source term.
- Combine first and second order displacements to give position and velocity.

Naively this can result requiring memory for nine or ten number fields. For a code like PKDGRAV3 where the size of a particle can be as low as 28 bytes this results in extra memory being required. It is possible to construct second order initial conditions using only six number fields by following this prescription which is based on the process in [171]. The ϕ terms are the partial derivatives of the density field calculated in Fourier space and transformed.

- (3) Construct the first-order terms as normal.
- (3+3) Construct the diagonal terms $\phi^{(1)}_{,00}$, $\phi^{(1)}_{,11}$, $\phi^{(1)}_{,22}$.
- (3+1) Calculate $\phi^{(1)}_{,00}\phi^{(1)}_{,11} + \phi^{(1)}_{,00}\phi^{(1)}_{,22} + \phi^{(1)}_{,22}\phi^{(1)}_{,11}$

- (3+2) Calculate $\phi^{(1)}_{,01}\phi^{(1)}_{,01}$ and subtract from previous.
- (3+2) Repeat for $\phi^{(1)}_{,02}\phi^{(1)}_{,02}$ and $\phi^{(1)}_{,12}\phi^{(1)}_{,12}$
- (3+1) Generate source term.
- (3+3) Calculate second-order displacements.

Chapter 4

Gravity Calculations

In this chapter we discuss the technique used to integrate an n-body simulation. The bulk of the work in PKDGRAV3 involved improving the code so it would scale to thousands of nodes.

4.1 PKDGRAV2

The starting point of this code improvement process was the excellent PKDGRAV2 code which is a direct descendent of the PKDGRAV[195] code. In this section we describe the various technology features of the code.

4.1.1 Force Hierarchy

As gravity operates over an infinite distance, it is technically necessary to calculate the force for each particle as a result of every other particle. Formally, this means that the gravitation potential is defined as:

$$V(x) = \sum_{i=1}^{n} -\frac{Gm_i}{x - x_i} \tag{4.1}$$

This scales as $O(n^2)$, so an exact solution is possible only for relatively small values of N, even with the very fastest computers. For this reason, a number of approximations are used while bounding the force errors to within a certain tolerance, suitable to the problem under consideration (more on this later).

For very close interactions, the force calculations are calculated directly as with the naive approach. Imagine a small sphere around the sink particle; all particles within this sphere are calculated directly.

As the distance is increased, it starts to make sense to treat groups of particles together as a "point mass". As a simple first step, a group of particles can be treated as as a point mass by calculating their centre of mass. If the sphere containing these particles is suitably far away, and suitably small then this approximation has acceptably low errors.

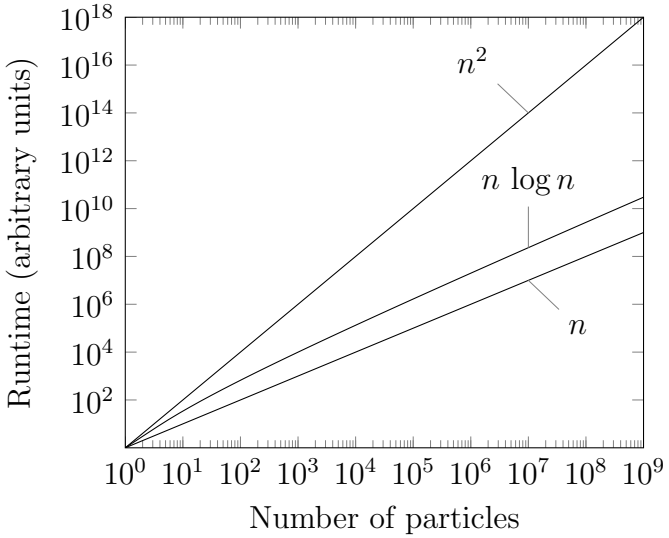


Figure 4.1: Scaling for different algorithmic complexities

The very first tree codes [14] stopped at this point and were very successful at the time. This approach can theoretically achieve a computational complexity of $O(n \log n)$, significantly better than the $O(n^2)$ achieved with the direct approach, but it is possible to do significantly better.

By increasing the order of the expansion from a monopole (zeroth order), to some higher expansion (quadrupole, octapole, etc.), it possible to move these cells close and make them larger. There are more calculations for each interaction, but there are far fewer interactions to calculate. For a given desired level of accuracy, an optimal order can be determined.

It should be noted that this does not change the order of the algorithm; it is still $O(n \log n)$, but it does change the shape of the scaling relation. It is actually slightly more expensive if the desired level of accuracy is very low. As the level of accuracy is increased, the run-time improves when compared to a first order approach, and very quickly surpasses it.

To change the order of the algorithm, a further improvement needs to be made, the Fast Multi-pole Method (FMM) [48]. It is possible to calculate the "shape" (multi-pole) moment of far-field forces and shift them to each particle where the forces are evaluated. This technique is $O(n)$! Figure 4.1 shows the scaling for the different algorithmic complexities. By 1 million particles (a very small number by today standards), a calculation that took 1 minute for an $O(n)$ algorithm would take close to 2 years for $O(n^2)$. As bad as this seems, it gets worse as N is increased. For a one billion particle halo, a calculation that takes a single **second** for an $O(n)$ code would now take 317 **years**! The order $O(n \log n)$ algorithm would take only 30 seconds.

It should finally be noted that for cosmological simulations there are one set of forces remaining. The size of the Universe is effectively infinite, but it is not possible to simulate an infinite volume at **any** resolution, the area outside the simulation area must be handled in a special way. The usual method is to create an infinite number of replicas of the simulation volume and use numerical techniques to approximate (or solve exactly in some cases) the forces that would result. This technique is known as periodic boundary conditions and is

implicit in spectral techniques (for example Tree-PM), but which is treated explicitly in pkdgrav2 by using Ewald summation [90].

Direct Interactions (Particle/Particle)

These are the easiest and most straightforward calculations, and they are evaluated as in equation 4.1. By using a number of numerical "tricks" it is possible to greatly accelerate the evaluation of these forces; tricks that will be covered in a subsequent section.

Multi-pole Approximation (Particle/Cell)

Once particles move beyond a certain distance, groups of them can be treated as a single multi-pole. The further away they are, the larger the collection of particles that can be approximated by a multi-pole. This is the technique that results in a $O(n \log n)$ algorithmic complexity, and formed the basis of the very first tree codes.

Local Expansions (Cell/Cell)

Particles outside an even greater radius can be approximated by a local expansion — the field that would be seen at the current particles position as a result of the other particles. In principle it is possible to use only particle/particle and cell/cell interactions, but using particle/cell for the intermediate range interactions results in better computational efficiency.

Periodic Boundaries (Ewald)

With cosmological simulations, the size of the simulation should be effectively "infinite", otherwise the lack of mass outside will affect the force calculations and the simulation volume will tend to collapse. This would be a purely numerical effect. The most common way of approaching this problem is to duplicate the simulation volume in all directions infinitely. Spectral methods (grid codes or TreePM) handle this implicitly by the very nature of the fast Fourier transform. In pkdgrav2 , it is handled explicitly using Ewald summation [90].

4.1.2 Control

Processor Tree

At the very start, all processors are arranged into a balanced binary tree like the one shown in figure 4.2. The left node is always local, while the right node is always remote. When the hybrid MPI/thread mode is in effect, the tree is built to a node, then continued down to the threads as in figure 4.3. In either case, each leaf in the processor tree is assigned a unique number from zero to the ones less than the number of domains. For the MPI-only situation, this number is the same as the MPI rank.

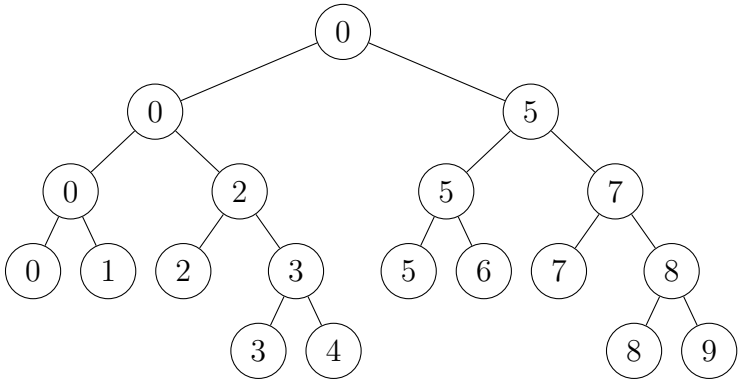


Figure 4.2: Processor tree with ten independent MPI domains.

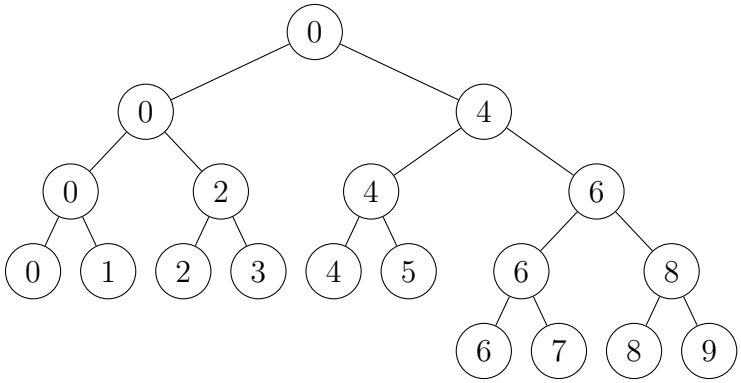
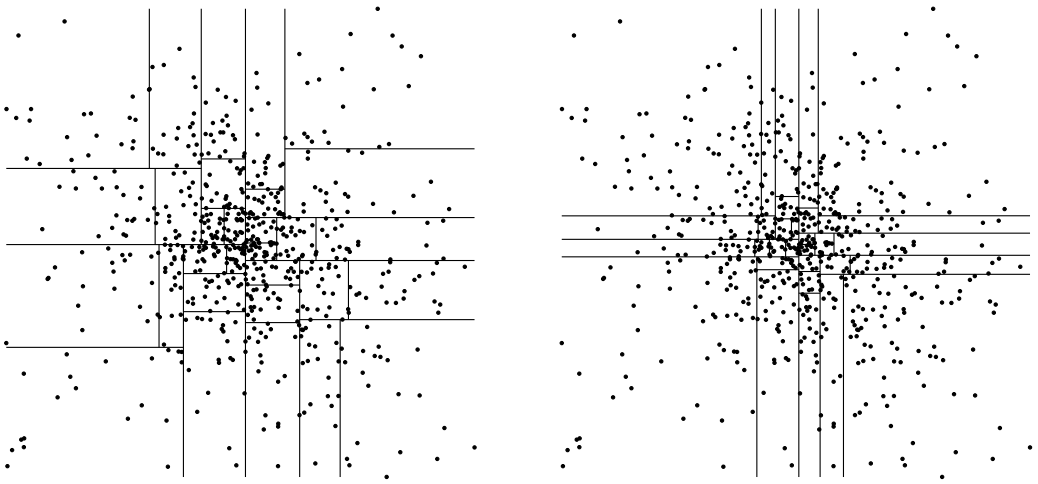


Figure 4.3: Processor tree with five nodes, each with two threads.



(a) Here the distribution of particles is divided into 15 domains, each with 21± 1 particles.

(b) Domains for 241 active particle only. Domains in the centre have as few as 8 particles.

Figure 4.5: ORB Domain decomposition.

4.1.3 Domain Decomposition

In a parallel codes, the work needs to be divided as evenly as possible between the available compute resources. with pkdgrav2 , the amount of work is related to the number of particles, and the total number of **active** particles turns out to be an excellent first-order estimate.

ORB

We use Orthogonal Recursive Bisection (ORB) [166] to divide the the particles spatially. In this scheme, space is divided into two regions by cutting along one of the dimensions (x,y or z) such that the number of particles on both sides are equal[1]. The process is repeated for the two halves, then for the four quarters and so on until the space has been divided into P different regions. The chosen dimension can (and usually is) different for each iteration.

The process is complicated somewhat as it is desirable to split up the total *work* evenly, not necessarily the number of particles. For simplicity, the amount of work for a given particles is zero if it is inactive, or one if it is active. This is a reasonable approximation with a fast multi-pole code, but it should be noted that the amount of work for inactive particles is not quite zero, and the amount of work for active particles is not exactly the same. Figure 4.5b shows a distribution of particles divided using a 2D ORB decomposition, both with all particles active, and for only a subset of particles.

From this figure it can be seen that as the number of active particles decreases, the balance of particles in a given domain tend to degrade. At some point, it is no longer possible, due to memory constraints, to fit all of the particles into some of the domains. At this point it is necessary to relax the strict "work" based approach, and allow the amount of work to vary between domains. This make the computations less efficient, as some of the processors will have completed their work before others, and are waiting.

[1]Or proportional to the number of domains on each side when the number of domains is not even.

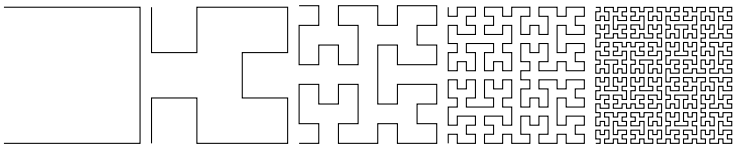


Figure 4.6: Hilbert Curves of different ranks.

Another problem with the ORB decomposition is that it tends to cut through high density regions. This has the effect of increasing the number of adjacent particles that are in different domains, and hence which are on different processors or even on different nodes. Fetching particle information from different domains is relatively expensive, and it would be better to minimize this as much as possible.

Another limitation is that the decomposition must be done in phases; the first cut must be calculated before the second which must be calculated before the third and so on. Each calculation requires a root find, and multiple synchronisation points.

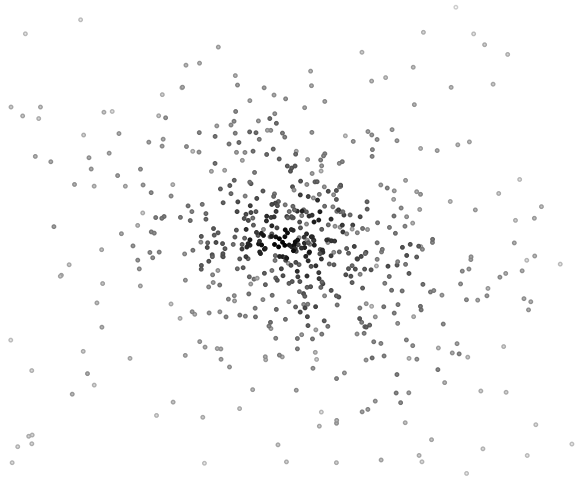


Figure 4.4: A distribution of 629 particles forming a halo.

Hilbert

A different domain decomposition method revolves around the Hilbert key. This is a one-dimension mapping of a three- (or higher) dimensional space. Each particle is assigned a hilbert key which follows a space-filling curve as shown in figure 4.6. The Hilbert is shown to fifth order, but can easily be computed for arbitrary accuracy. The space filling curve is then split into P different segments, each with an equal share of work (using whatever metric desired). The number of iterations and synchronisation points for this process is similar to a single cut in the ORB method, so the time required is largely independent of the number of domains.

This method also has improved locality over the ORB decomposition; in fact it has been shown that the surface area is minimized over all ensembles. The down-side to this approach is the rather awkward domain shapes that end up resulting in extra communication between nodes.

4.1.4 Tree Format

The multi-pole approximation technique relies on the ability to approximate areas on space as ever increasing larger multi-poles as the distance from the particle increases. An efficient data structure for achieving this is required, as is traditionally a tree. In [14], an Octree was used, but this is not the only option. In pkdgrav2 , a binary tree is used instead.

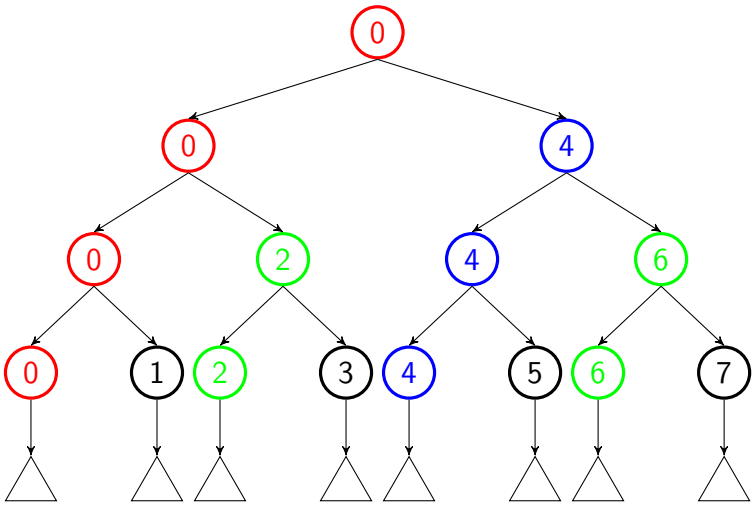


Figure 4.8: Top Tree. A node will have at most $\log_2 N$ cells. In this example node 0 has 4 cells, node 4 has 3 cells, nodes 2 and 6 have 2 cells and nodes 1,3,5 and 7 have one cell.

With an Octree, when a cell must be divided, it is split into eight children cells, all of the same size. This has the advantage of keeping the cells all the same shape — their length, width and height are all identical. This simplifies the acceptance criteria as the geometrical "shape" of the cell (and hence it's corresponding multi-pole) are more uniform in all directions. This is not without a cost. At the bottom of the tree, there are multiple cells that are empty which results in unnecessary storage use, and computation (see figure 4.7).

In pkdgrav2 we use a binary tree instead. Cells are split iteratively as necessary resulting in cells that are shaped like a cube, slab, a pencil, and then back to a cube, and so on.

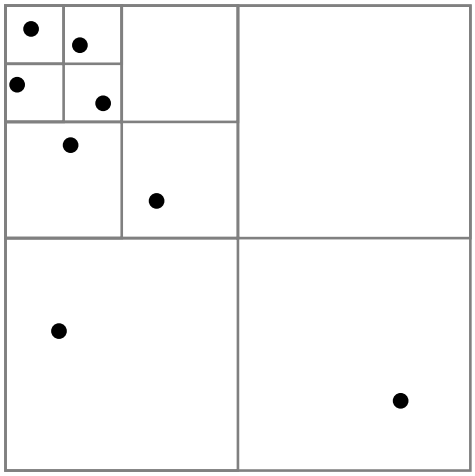


Figure 4.7: Octree construction

4.1.5 Top Tree

The tree is continued from each node upward via the PST.

4.1.6 Tree Walk

During the tree walk, we will distinguish between two kinds of cells; sink cells which receive the force, and source cells, the source of the forces. To kick-start the process, we start with the root cell (the entire simulation volume) as the first sink cell. It is also the first source cell as we want to apply the force to to all other particles to all particles. The source cells go onto a list called the "check list", while the sink cell goes onto a stack of cells that still need to be evaluated. If periodic boundaries are in effect, the the Ewald replicas are added to the check list right at the beginning (more on this later).

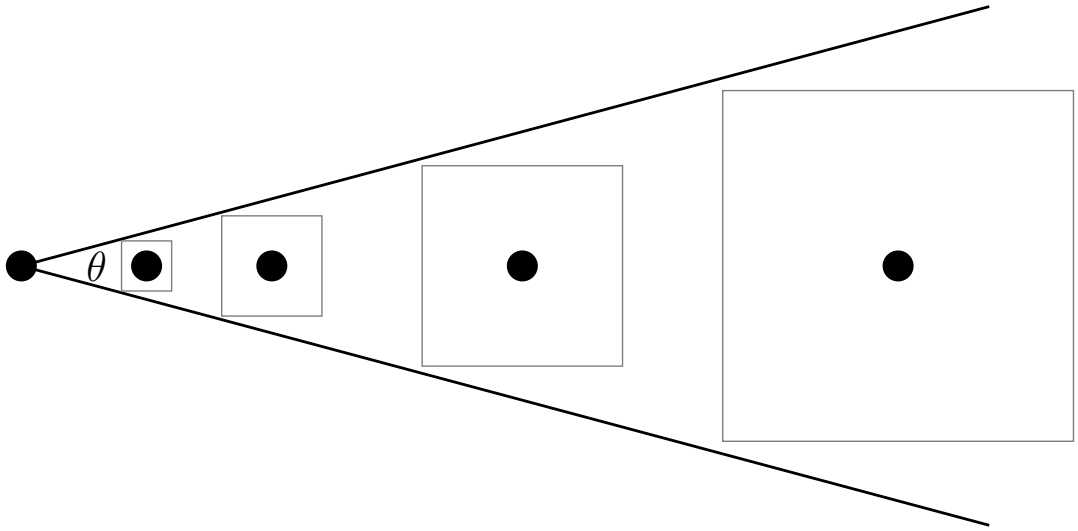


Figure 4.9: Opening Angle. The more distance the tree cell, the larger it can be and still be accepted as a multi-pole[195].

The tree walk proceeds by comparing the current sink cell with all of the source cells on the check list, and determining how the force interactions should be evaluated. Obviously, at the very start the two cells overlap (they are the same cell) so it is impossible to evaluate the forces. Instead, one of the cells must be "opened" so that their two child cells can be separately considered. If a source cell is opened, then its child cells are added back onto the check list. The check list will continue to grow until the cells are sufficiently small for their forces to be evaluated in one of several different ways.

When a sink cell needs to be opened, one of the child cells is pushed onto a stack, along with the current check list. Once a sink cell has been completely evaluated, the next cell on the stack will be retrieved and processing will continue.

When the stack has been emptied, then all force evaluations will be completed. The stack can only be as large as the deepest part of the tree which is roughly $\log_2 n$; n being the number of particles local to this processor.

4.1.7 Opening Criteria

During the tree walk, pairs of cells are compared to determine how their force interactions should be evaluated. The first cell, the "'sink" cell will receive the forces evaluations while the second, or "source" cell will provide them. With pairs of cells, the following outcomes are possible.

1. The two cells are too close, so the larger of the two is "opened". If this is the source cell, both child cells are added to the checklist, and processing begins with the next cell on the checklist.

2. If the sink cell is the larger of the two, then one of the child cells is pushed on the stack, along with a copy of the current checklist, and processing continues with the other child cell.

3. If the source cell is sufficiently close and has sufficiently few particles, these particles are applied as direct particle/particle interactions.

4. If the source cell is sufficiently far away, then it can be treated as a local expansion; it's multi-pole moment is combined with the multi-pole moment from other source cells, shifted to the position of the sink cell, and evaluated there.

5. If the cell is sufficiently small, but not far enough away to be treated as a local expansion, it can be evaluated as a particle/cell interaction.

The basis of how these decision are made is through an opening angle (see figure 4.9).

4.1.8 Time Integration

Having calculated the forces acting on a particle, the next step is to integrate it forward in time. There exists a plethora of work on this exact problem and a number of different integration schemes exists, each with its own advantages and disadvantages.

4.1.9 Multi-Stepping

It turns out that different particles in the simulation require fewer or more steps to accurately integrate their orbits; a good rule of thumb is that a body should take of order 30 steps for each orbit. The most common way of determining this time is the so called Epsilon-over-a criteria where the time step ΔT goes as:

$$\Delta T = \eta\sqrt{\frac{\epsilon}{|a|}} \tag{4.2}$$

where η is the time-step parameter, ϵ is the particle softening, and $|a|$ is the magnitude of the acceleration.

4.2 PKDGRAV3

4.2.1 Algorithmic Changes

Mixed Precision Floating Point

Numerical simulations have traditionally been done using double precision floating point numbers. There are very good reasons for this: double precision provides around 15–17 digits of precision while single precision provides only 6–9. The GHALO simulation [196] for example, has a softening length of 61 pc in a 40 Mpc volume. This corresponds to 10^{-6} which is already at the limit of the precision. For this reason, phase space coordinates (e.g., positions and velocities) must be kept in double precision.

At the same time, it would be nice to take advantage of the fact that modern CPUs can perform twice as many single precision operations as double precision in the same time (effectively doubling performance).

Another related problem is with the range of exponent which can be problematic with high order schemes as the high order terms can underflow to zero. For example, the GHALO simulation [196] has a softening length of 61 pc in a 40 Mpc volume. This corresponds to 1.5×10^{-6} which seems far away from the float limits (approximately 10^{-44}), but as pkdgrav2 uses a high-order multipole expansion, this can quickly cause problems. To combat this problem, the multipoles are scaled by their magnitude, so that each term is of order unity.

Integerized Coordinates

As described in the previous section, the precision of single precision float positions is insufficient, not because of the total number of bits available, but because of the size of the mantissa. By integerizing the positions, it is possible to store them using a single 32-bit integer and still have sufficient resolution. Integerized coordinates have a range of between $\pm2^{31}$ which corresponds to a grid spacing of smaller than 10^{-9}. This is at least three orders of magnitude below the softening length of even the very highest resolution simulation.

Integerizing the coordinates has the effect of reducing the storage required for the positions from three double precision float values (24 bytes) to three 32-bit integers (12 bytes) allowing even larger simulations to be run in the same amount of memory.

Hybrid MPI/thread model

On of the major changes to PKDGRAV3 was the switch from a pure MPI **or** pure threads model to a hybrid model that uses both MPI and threads at the same time. The motivation was two fold: first, there is an underlying overhead when using MPI that scales weakly with the total number of MPI processes. Instead of a single MPI process per *core* we use a single MPI process per *node*. This results in a dramatic decrease in the number of MPI processes; on Titan for example, we would go from 270,000 processes down to 18,000 when using 18,000 nodes.

The second motivation is the performance increase gained by using a threading model. With MPI, all communication must be done using message passing, even to acquire information that is on the same node. By using threads, the overhead of message passing can be eliminated in most cases for data exchange on-node. As the number of cores per node increases, both of these effects become more and more important.

This introduces additional complexity. Most MPI implementations do not play well with multi-threaded applications. Either the semantic is not supported at all, or additional delays are introduced as it then becomes necessary to perform locking to prevent race conditions. To avoid this, all communication is channelled through a single MPI thread. Cache requests and flushes are bundled on each local thread before being sent the the MPI thread. The MPI thread then unpacks the messages from all local threads and repackages them based on the target node. For messages received from other nodes, the process is reversed: message are unpacked and bundled for delivery to the individual threads. This bundling process is

essential, otherwise the overall performance is greatly degraded. Data that exists locally on other threads can be fetched immediatly, bypassing this process entirely.

Latency

The underlying parallization run-time paradigm uses a "fetch on demand" semantic for many operations. When walking the gravity tree for example, nodes and particles from other domains (both on remote nodes and owned by other local threads) are needed. Thus, the latency becomes extremely important (i.e., the time between requesting data and receiving it) because the algorithm cannot continue until the data arrives. As data access is quite local, a caching mechanism is implemented where several values are fetched for any cache miss with the (correct) understanding that nearby particles or tree nodes will also be needed shortly.

For actual cache misses, the entire pipeline has been optimized to minimize latency. On a cache miss, a request is sent to the MPI thread using "lockless" communication. As the MPI process is dedicated to the task, it immediately receives the message and transmits it to the MPI process on the remote node. The MPI is also waiting for incoming message and immediately sends the response. For background operations, such as cache flushes, the data is buffered and delivered to threads to process at their leisure.

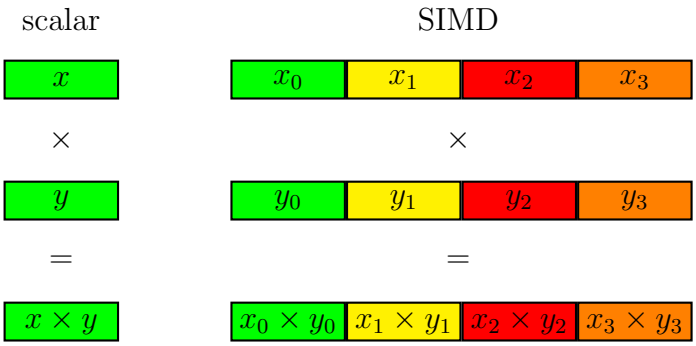


Figure 4.10: Scalar versus SIMD

x_1	y_1	z_1	m_1	ϵ_1
x_2	y_2	z_2	m_2	ϵ_2
x_3	y_3	z_3	m_3	ϵ_3
x_n	y_n	z_n	m_n	ϵ_n

(a) Optimal particle layout.

x_1	y_1	z_1	m_1	ϵ_1
x_2	y_2	z_2	m_2	ϵ_2
x_3	y_3	z_3	m_3	ϵ_3
x_n	y_n	z_n	m_n	ϵ_n

(b) Optimal SIMD layout.

Figure 4.11: Conflicting particle layouts.

4.2.2 Hardware Acceleration

SIMD

One of the more recent developments is the move towards data parallel operations on commodity hardware. Intel now dominates the CPU market and they have implemented Single Instruction, Multiple Data (SIMD) instructions. Traditionally, each instruction would perform a single arithmetic operations, for example multiplying two numbers. With SIMD it is possible to perform multiple arithmetic operations in the same amount of time; four at a time for SSE, and eight at a time for AVX. The limitation is that the operations must be identical and only the data can be different. This is shown in figure 4.10.

The data format and alignment must also obey certain restrictions. The elements must be physically adjacent in memory, and they must be appropriately aligned[2].

Internally, particles are arranged such that all information about a particular particle is adjacent in memory as shown in figure 4.11a. This is generally good as the CPU will prefetch all particle data when a particle is first accessed. However, this is not a good layout for SIMD operations. A better layout is to have all of the x coordinates together, all of the y coordinates together, and so forth as shown in figure 4.11b.

[2]Exact alignment requirements can vary between architectures.

Particle/Particle Interactions

The direct interactions can be done in single precision, effectively resulting in a factor of 4 (SSE) or 8 (AVX) increase in performance. Positions cannot be used directly as they need to remain in double precision or the results will not be accurate enough. To accomplish this, a reference point is chosen that is near the particles being considered. This is the coordinate of a particle in the current bucket. When particles are placed on the interaction list, their position is re-centred relative to this coordinate. When a new bucket is being considered, any particles on the interaction list are adjusted to the new centre. Because of the short range nature of the direct interactions, the adjustments to the coordinate system result in virtually no errors as particles will leave the interaction list before they get far enough away for it to become a factor.

Particle/Cell Interactions

Cell Opening Criteria

As described in section 4.1.6, the cell opening criteria is rather complex computationally, and results in one of the following outcomes.

- The cell remains on the interaction list.
- The cells particles are treated as direct (particle/particle) interactions.
- The particles in a cell are added back on the checklist as "cells".
- The cell is opened, and the child cells are put on the checklist.
- The cell's multi-pole expansion can be accepted (particle/cell interaction).
- The local expansion can be accepted.

Ordinarily, the fact that there are multiple possible outcomes would indicate that the use of SIMD instructions is not possible, or at least not feasible as SIMD instructions are vector operations, and vector instructions must perform identical operations on each operand. As a side effect, conditional branches are not possible which makes the logical if/then/else nature of the criteria impossible.

As it turns out, there is a very good alternative. Most, but not all of the heavy computations for calculating the opening criteria need to be performed regardless of the eventual outcome. For example, it is always necessary to calculate the distance between the two cells. Additionally, there are SIMD instructions to perform comparisons of vectors which result in words of all ones if the result of the comparison is true for a given element, or all zeros if the result is false. By combining these comparison instructions with logical bit manipulations (and, or and not), it is possible to calculate, in parallel, the result of the opening criteria. The resulting speedup is very close to the width of the SIMD instructions (four for SSE and eight for AVX) for two simple reasons.

1. We can calculate four (or eight) at a time, with only a few additional, unnecessary instructions, and,

2. We remove all branches from the calculations.

Item (2) is important because branches on modern CPUs can result in "stalls"; the processor must make a decision as to if it will fetch the next instruction after the comparison, or if it should fetch the instruction at the target of the branch. If it guesses[3] wrongly then it must clear the pipeline and fetch the correct instruction. This is quite expensive. To make matters worse, the nature of the opening criteria makes the choice of outcomes effectively random enough such that branch prediction algorithms will only be correct half of the time.

Ewald

The improvements to the other parts of the gravity calculation had the predictable effect of making Ewald become a more significant component. Fortunately it is also possible to accelerate the calculations using SIMD. There are three components to the Ewald force, and they are treated differently.

1. The particle is within a small sphere in the center of the main box.
2. The particle is outside this sphere, but within the number of replicas.
3. The particle is outside the periodic replicas.

There is one main box, and 26 (nRep=1) or 124 replicas (nRep=2). We treat the main box as before with the forces evaluated using efc, or a series expansion if we are in the very inner sphere to avoid errors caused by large cancellations.

The replicas are calculated by using SIMD parallel operations which reduces the time by a factor of 2 (SSE) or 4 (AVX). These calculation are done in double precision as single precision does not give sufficiently accurate results.

The correction to the forces outside the periodic replicas (the so called h-loop) can be done with single precision SIMD instructions. The overall result is that the Ewald calculations again become only a tiny fraction of the total work.

4.2.3 GPU

Another recent development in high performance computing is the addition of hardware accelerators, specifically nVidia GPU compute devices. These devices provide a very high theoretical number of floating point operations per second when compared to general purpose CPUs, and are thus of great interest for scientific computing. A modern 12-core CPU is capable of achieving half a teraflop for 130 Watts while an nVidia GPU can theoretically achieve five teraflops for 250 Watts. Not only does the GPU give better total performance, but the performance per Watt is also much better; 20 GFLOPS per Watt for a GPU device versus 4.5 for a CPU.

While it is somewhat of a challenge to achieve near-peak performance for a CPU, it is extremely difficult for a GPU based device. The tools and technology are reasonably new

[3] "predict" is the usual word used here.

and hence not very mature. In addition, the restrictions for a GPU code are much more confining, and can have a dramatic impact on performance.

There are two basic approaches to GPU programming. In one approach, all of the data is copied to the GPU, and all calculations are performed there. This is a good approach if the problem can fit on a single GPU, or if the time to transfer the data to the GPU is sufficiently shorter than the computation time.

In the case of pkdgrav2 , the problem generally will not fit in a GPU; a typical GPU has several gigabytes (between three and six), while a CPU will normally have many tens or hundreds of gigabytes of RAM. In astrophysics, the amount of RAM is often one of the limiting factors so restricting to the problem to fit into one or more GPUs is not an appealing option.

The other approach, and the one used in pkdgrav2 , is to treat the GPU as an accelerator. Units of work are transferred to the GPU where the calculations are performed, and the results are transferred back to the GPU. The other aspect that makes this necessary is the long-range nature of gravitational forces. All nodes in a cluster need accurate and up-to-date particle positions from other nodes, so all of the information needs to be accessible to the CPU, so transferring between the CPU and GPU is unavoidable.

The limiting factor with GPU acceleration is the so-called flop to byte ratio; the number of floating point operations that need to be performed for each byte transferred. If this number is high enough, then the total transfer time will decrease with respect to the computation time. At some point, the transfer time will equal to computation time, and it is possible to overlap the communication and computation making the communication effectively disappear. It is thus highly desirable to have a high flop to byte ratio. This is usually achieved by data reuse, rather than having an actual large amount of computation, though that can also be the case.

Particle/Particle (P-P) Interactions

This is one of the more challenging force calculations to accelerate with the GPU as the float to byte ratio is quite low. There are only five quantities that are necessary to perform the force calculation: position (x,y,z), mass, and force softening. The number of floating point operations is very low: on the order of ten to twenty for unsoftened interactions and twice that for softening interactions. This is too low a ratio to get efficient use of the GPU, but there are ways of improving this.

The first, and obvious improvement comes from the tree walk algorithm. For P-P interactions we always apply a bucket of particles to a **group** of particles instead of to a single particle. We can tune the size of a group, but 64 is already optimal for CPU interactions and with that we get a roughly factor for 40 improvement in the flop to byte ratio – each particle in the group needs to calculate the forces on all particles on the interaction list.

Another huge win is that interaction units are usually “groups”, or “buckets”. Knowing this, we transfer entire groups with their corresponding buckets to the GPU whenever a single particle is needed. We can then send groups and interaction lists as a group identifier (four bytes), or as a list of buckets (four bytes instead of approximately 200). As we have

significant reuse (a bucket will need to interact with many buckets in its neighbourhood), we can reduce the communication requirements by an order of magnitude.

To optimize memory access, buckets are fixed to the size of a warp. We know that this does not adversely affect the accuracy, and, at the current warp size of 32 the performance is in the optimal range. Buckets are then paired to form a complete warp size; 31 with 1, 30 with 2, and so on. When interaction lists are sent to the GPU, they are similarly paired, although the pair may be different. This allows the buckets to be efficiently fetched from GPU global memory without bank conflicts.

As the size of the GPU memory is typically an order of magnitude smaller than the CPU (3 GB versus 64 GB for example), it may not be possible to fit all buckets into the GPU memory at the same time. The buckets on the GPU are treated as a cache, and sent as they are needed. The local nature of the tree walk guarantees high cache reuse, and in most cases a bucket will only need to be sent once.

The overhead associated with starting a GPU transfer is also quite significant, so to ameliorate this effect, both buckets and interaction lists are held on the GPU until a suitably large number of them exist to begin a data transfer. This size is chosen such that the transfer start-up time becomes insignificant. This cannot be made arbitrarily large as then the latency will start to become an effect.

Particle/Cell (P-C) Interactions

With P-C interactions, it is not possible to use the same trick as with P-P interactions where large sets of interactions are grouped together. A single bucket of particles will have the same set of interacting cells, but these interacting cells cannot be easily grouped together (as with "groups" in P-P interactions).

The good news is that cells are "fat" in that they contain a high-order multi-pole – they contain 26 coordinates instead of the five of a particle. In a sense then, each cell is similar in "weight" to a group of particles.

Similar to the P-P interaction, P-C interaction lists are sent as a list of cells (rather than individual values), and the cells are cached on the GPU. This allows a cell of roughly 100 bytes to be sent to the GPU once to the cache, and then as a four-byte identifier on interaction lists. As cells appear on the interaction list of multiple buckets, this decreases the transfer size by an order of magnitude.

Periodic Boundary (Ewald)

The calculation of the periodic boundary conditions using Ewald is ideally suited to offloading to the GPU. The calculations require a single (shared) multi-pole that represents the unit box, and the position of each particle. This results in tens of thousands of flops for each particle. At the moment, double precision is required to attain the necessary level of accuracy so only certain GPU devices (most notably the Tesla line) are suitable for this offloading.

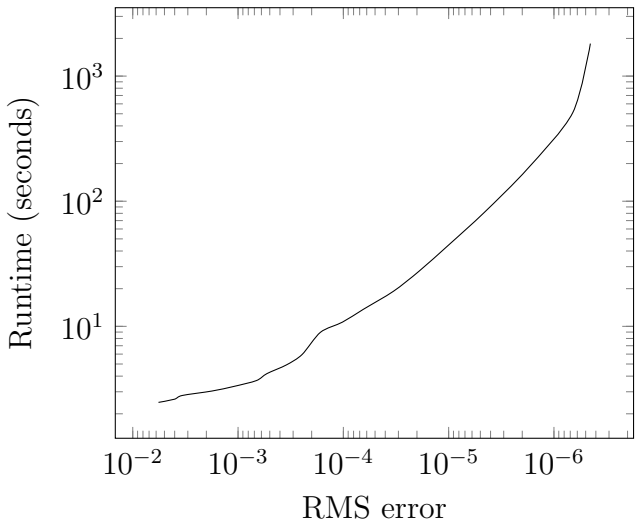
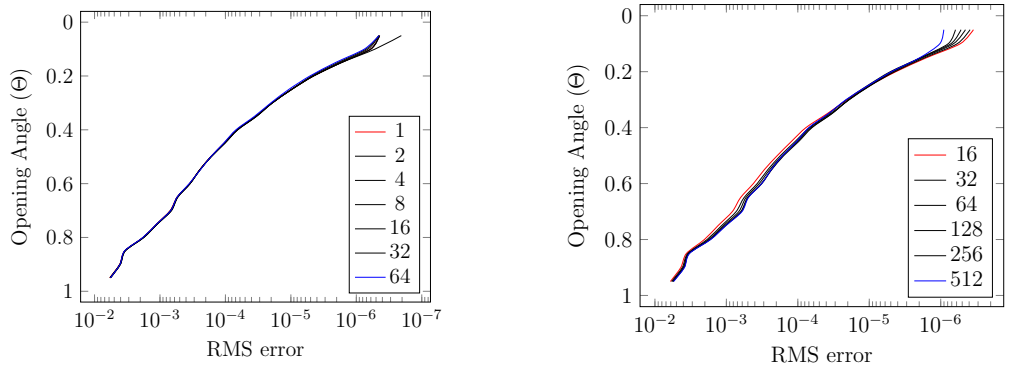


Figure 4.12: Force accuracy versus time



(a) Accuracy for different bucket sizes.

(b) Accuracy for different group sizes.

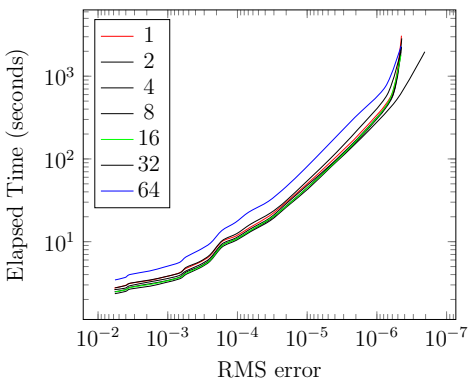
Figure 4.13: Accuracy for choice of group/bucket sizes.

4.2.4 Accuracy

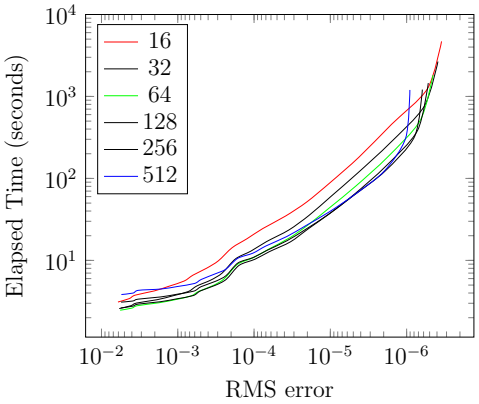
None of these performance enhancements are of any value if they have an adverse affect on the simulation accuracy. To test the accuracy of the force calculations, we can compare the results to an identical calculation done directly without a tree code. The simulation must be reasonably small as the direct force calculation scales as $O(n^2)$. In this direct code, all force calculations are done in double-precision floating point, so the forces are accurate to within approximately 10^{-15}.

The results are shown in figure 4.12. For cosmological simulations, the force accuracy is normally calibrated to 0.1%. This level of accuracy is easily reached with a modest opening angle. It is possible to achieve RMS errors of better than 10^{-6} which is the accuracy limit for single precision floating point numbers. To achieve higher precision it would be necessary to move away from a fixed precision model in favour of double precision for all calculations. This would result in a factor of two decrease in performance on the CPU, and a much larger decrease in performance on the GPU.

The accuracy is unaffected by the choice of bucket size (figure figure 4.13a), and largely unaffected by the choice of group size (figure 4.13b). Increasing the group size results



(a) Science Rate for different bucket sizes.



(b) Science Rate for different group sizes.

Figure 4.14: Science Rate for choice of group/bucket sizes.

in a corresponding increase in the number of direct interactions (particl/particle), and so achieves a slightly higher accuracy with the same opening angle. This comes at a slight cost, so a better measure is the "'science rate".

As can be seen in figure 4.14, the situation is somewhat changed. There is an optimal bucket size — 64 is too large to be efficient while 1 is unnecessarily wasteful of memory (the tree size is affected by the bucket size). The ideal number seems to be either 16 or 32 which fits well with the CUDA/GPU strategy. Similarly, the optimal group size is 64.

4.2.5 Performance

With the various changes that resulted in PKDGRAV3, it is now possible to run simulations of unprecedented size, including the Euclid Flagship Simulation with 2 trillion particles. Figure 4.15 shows the "time-to-solution" for the exact same simulation using PKDGRAV3, GADGET3 and RAMSES. The PKDGRAV3 code is at least an order of magnitude faster while still producing the same result (see chapter 8 for more details).

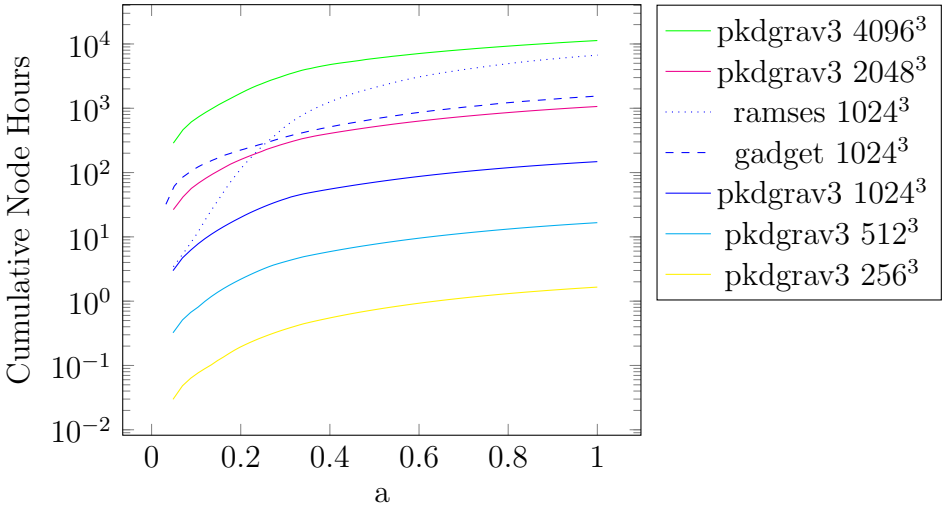


Figure 4.15: Speed comparison of PKDGRAV3, GADGET and RAMSES. Note that PKDGRAV3 is at least an order of magnitude faster.

Chapter 5

Structure finding

Once we can successfully run a simulation, the really important work begins. The simulation process results only in the position and velocities of the particles at different times in their evolution. In order to answer the important questions, it is necessary to somehow extract the useful information from this "mass of dots".

One of the most important tasks in this *analysis* phase is to identify structure; we often refer to this structure as "haloes", or "subhaloes" and their properties, including their distribution is of particular interest. In this section we present a high performance method of quickly identifying substructure in a simulation.

This method, called grasshopper , has been implemented into the simulation code pkdgrav2 which allows the analysis of the simulation to be performed during the simulation process. This is important as storing the properties of haloes takes orders of magnitude less storage when compared to the raw particle data. This is important as it means that it is possible to have better time resolution (i.e., more outputs) for large simulations where it is not practical to keep the necessary number of complete outputs.

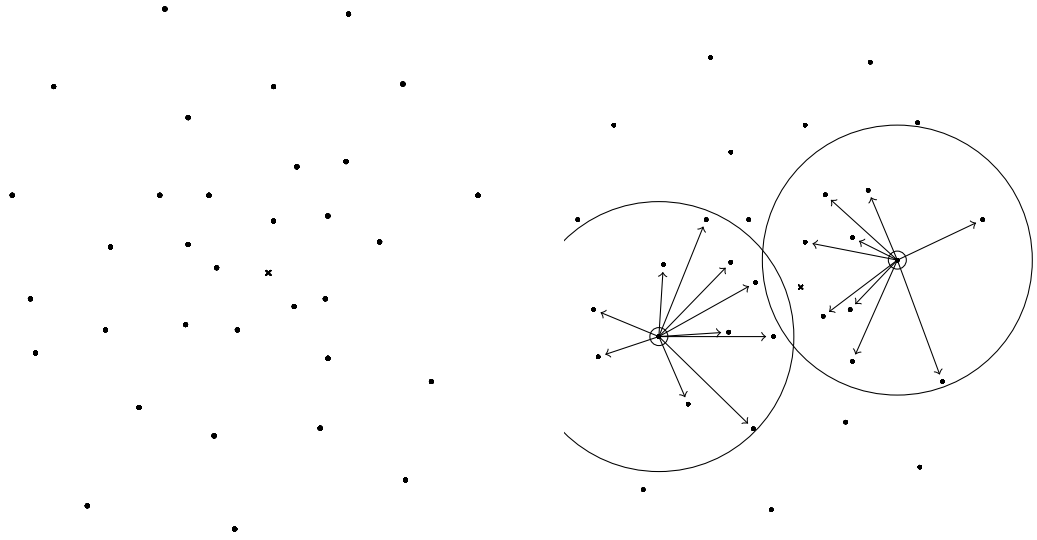
5.1 Prior Algorithms

There has been a recent resurgence of interest in halo finders in the last few years [107, 109, 144]. It is interesting to review a selection of these that have influenced, or which have algorithmic similarities to grasshopper .

In figure 5.1a an example distribution of particles can be seen. The idea is to somehow identify these particles as all being part of the same halo, or subhalo, or both. These halo finders all attempt to solve the same problem, each in a slightly different way.

5.1.1 Friends of Friends

The friends of friends algorithm [47] is one of the more widely used and understood group finding algorithms. This is one of the simpliest group finders that uses a linking technique, and one of the first group finders generally, so it is useful to review the process.



(a) A distribution of particles with the density peak indicated by "×". The goal is to properly identify these particles as being part of the same halo.

(b) The small circles indicate particles being considered. The large circles are the linking lengths.

Figure 5.1: Friends of Friends

The algorithm takes a single parameter; a characteristic linking length. For cosmological simulation, one fifth of the mean interparticle separation turns out to be a reasonable value as it corresponds to overdensities greater than 125 times the critical value. This is physically motivated as this is the overdensity at the virial radius [77, 74]. Every particle that is within this distance is linked together. Figure 5.1b shows this process. For a specific particle (small circle), all other particles within the linking length (large circle) are linked together.

The basic friends-of-friends algorithm suffers from a few problems. The choice of linking length is critical. If it is too large, then different halos will be linked together. If it is too small, then it will break halos into high density clumps. This behaviour has been exploited in other implementations [73] to identify substructure.

Another serious limitation is when the density increases, the complexity of the algorithm can move from $O(N)$ towards $O(N^2)$.

5.1.2 DENMAX

Recognizing the significant shortcomings of the friends-of-friends algorithm, Bertschinger and Gelb [25] devised a new scheme. In this scheme, particles are moved through a grid in the direction of the density gradient, until they pool at a local maximum. By using a fixed grid with a length scale appropriate for the simulation it is possible to identify substructure accurately while avoiding falls density peaks caused by noise.

One of the main limitations of this approach is that the length scale is determined by the size of the grid. If the grid cell is too large, then substructure is missed. It may not be possible because of memory constraints to construct a grid small enough to identify all substructure, particularly with "zoom" simulations.

5.1.3 SKID

The SKID [195, 74] algorithm was designed to overcome some of the limitations of the DENMAX algorithm. Instead of using a fixed grid to calculate densities and density gradients, an SPH approach was used. For each particle, the density is calculated with a scatter/gather approach, and is given by [195, 91]:

$$\rho(\mathbf{r}_i) = \sum_{j=1}^{N} m_j \frac{1}{2} \left[W(r_{ij}, h_i) + W(r_{ij}, h_j) \right] \tag{5.1}$$

and the density gradient by:

$$\nabla\rho(\mathbf{r}) = \sum_{j=1}^{N} m_j \nabla W(\mathbf{r} - \mathbf{r}_j, h_j) \tag{5.2}$$

First, the density of all particles is calculated. Next, a tracer particle is created for all particles. The density gradient is evaluated for each tracer particle and the tracer particles are moved a small distance in this direction. The process is repeated until all of the tracer particles have reached a density maximum. Particles that have pooled together at a given density maximum are said to form a group.

There are two critical parameters that control how SKID works internally. The first is the number of neighbours used to calculate the densities and density gradients. Increasing the number of neighbours tends to decrease the noise, and improve the estimates for density and the density gradient, but it also decreases the effective resolution. Emperically, 32 neighbours works well. It can easily been seen that by choosing a fixed number of neighbours, the algorithm is adaptive in resolution. In high density regions, the kernel is smaller and so smaller substructure can be identified.

The second crucial parameter is the length particles are moved during each iteration. This parameter must be small enough so that particles do not "jump over" density maximum. Making it too small increases the run time as more iterations need to be performed. Generally this is set to several times the softening length.

5.1.4 HOP

HOP [64] is another derivative of the basic DENMAX algorithm, and like SKID it avoids using a mesh approach in favour of a particle-only SPH method. However, unlike SKID, it does not use density gradients, but rather looks at each particles nearest neighbours and associates each particle with the neighbour with the highest density. This process is done for each particle and results in a chain of particles that terminate in a particle that is linked to itself.

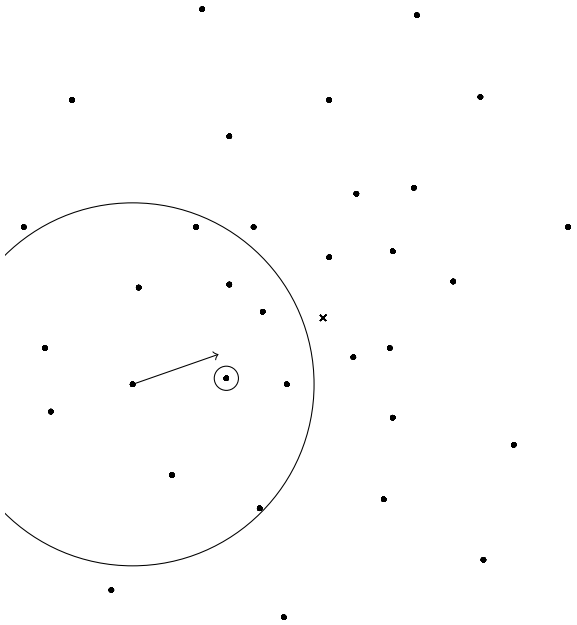


Figure 5.2: The smoothing kernel (circle), and a vector half this distance in the direction of the density gradient. The particle would be linked to the one that is circled as it is the closest.

5.2 grasshopper Algorithm

The grasshopper algorithm uses a technique similar to SKID, but is orders of magnitude faster. The basic idea is that instead of sliding particles slowly along the density gradient, particles are linked to the next particle along the density gradient. This results in one linking operation for every particle instead of many "slide" operations.

The density is calculated in the same way as in SKID using the SPH kernel in equation 5.1. From this density, the gradient is again calculated in the same way as with SKID. At this point the process differs.

5.2.1 Density and Gradient Calculation

The density calculation is calculated as with SKID using a SPH smoothing kernel. This calculation is completed before the density gradient is calculated, because the density of all neighbour particles is required for the gradient kernel. The second calculation can be slightly accelerated by saving the search radius (the bounding ball) from the density calculation as it will be the same if the same number of neighbours is used for both kernels.

During the gradient calculation, it is possible to perform the particle linking step described below instead of in a separate phase. This is done so that the gradients do not need to be saved (and consume memory), as only the particle link is ultimately required for the algorithm.

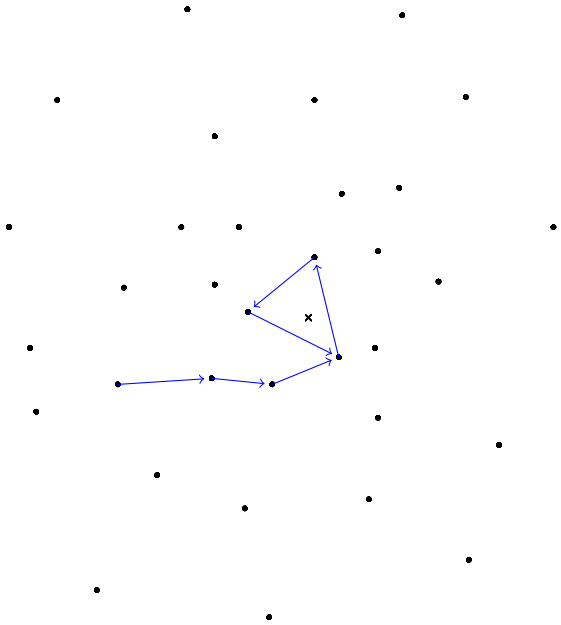


Figure 5.3: The chain of particles is followed until a particle that has already been linked is found. This will form a terminating loop.

5.2.2 Linking Particles in Chains

The basic density and gradient calculation processes identify a bounding sphere that contains the N nearest neighbours, and the direction of the density peak. A point is chosen that is a distance of half the radius of this sphere in the direction of the density gradient. The current particle is then linked to the particle closest to this point. This is illustrated in the figure 5.2. In the pathological case, the particle can link to itself.

The process is repeated for the next particle in the chain, until we reach a particle that has already been assigned a link. The first time that this happens, it will be the terminating loop of a new halo as shown in figure 5.3. This will form a small loop in the area of a density maximum.

We then start with another particle that has yet been assigned to a chain. This will again result in a chain of particles that ends in a loop, or it will result in linking to an existing group as shown in figure 5.4. In this figure we see two examples; in the first, a spur has been joined at the terminating loop, and in the second, the spur was attached to another spur.

Once all particles have been processed, they will all be linked to their own halo as shown in figure 5.5. All particles that are linked together form a halo, and in a cosmological simulation, there will be a halo for every over-dense region which results in the identification of all subhaloes.

5.2.3 Joining Chains

The result in figure 5.5 is an oversimplification, and the actual result will appear more like in figure 5.6. Because of noise, it is quite common for several chains to terminate in a density maximum, but to not be connected to each other. The termination of these chains

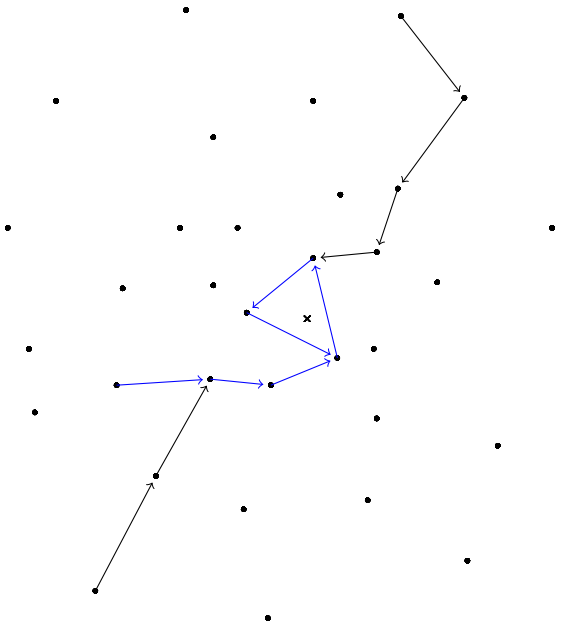


Figure 5.4: Spurs are joined to an existing chain by choosing an arbitrary point and following the density gradient until an existing chain is found. The spurs then become part of the existing chain.

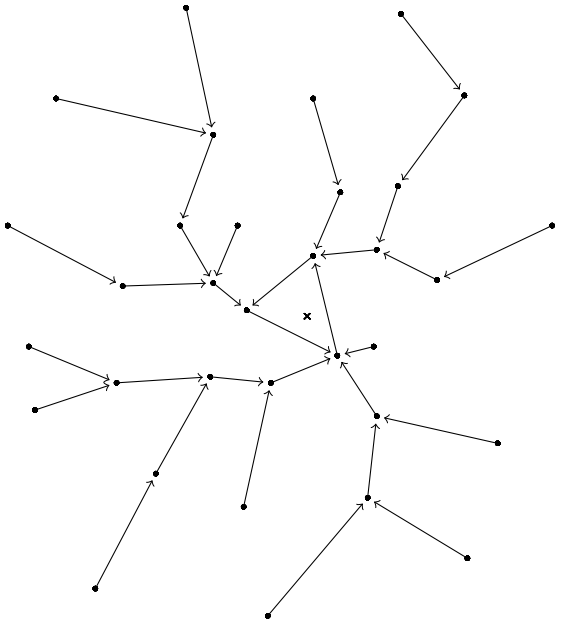


Figure 5.5: A halo that has been totally linked together.

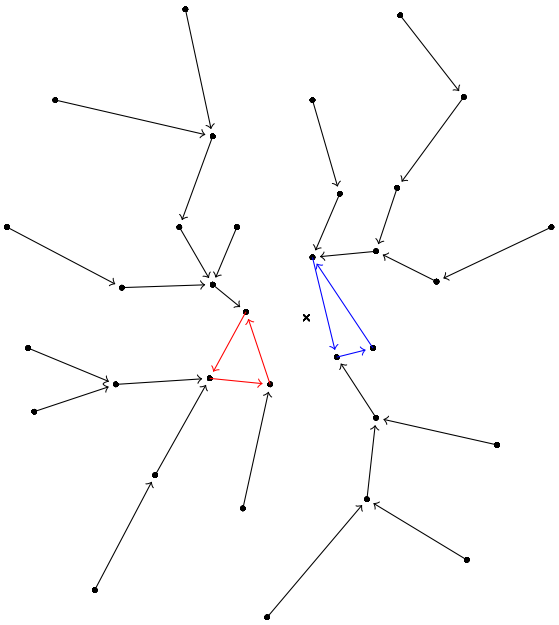
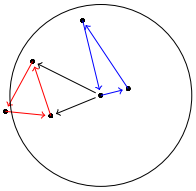


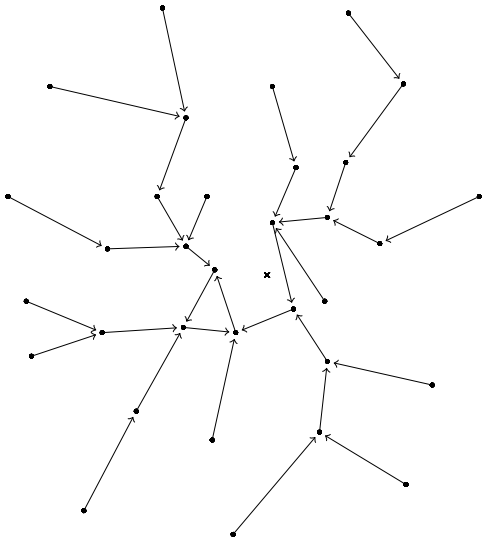
Figure 5.6: A single halo with two different terminating loops. These must be merged together.

will be very close together, so another algorithm is used to identify and link them to each other.

All particles that are part of terminating loops are separated from all other particles and a ball search is performed (figure 5.7). Any loops that fall within the ball search radius are linked to each other to form a new, larger group. The choice of ball search radius is a free parameter, but setting it to several softening lengths (e.g., 4) works well.



(a) Here we have removed all but the particles in the terminating loops. A ball search is performed to identify all particles that should be joined together (similar to friends-of-friends).



(b) The two chains have been merged together.

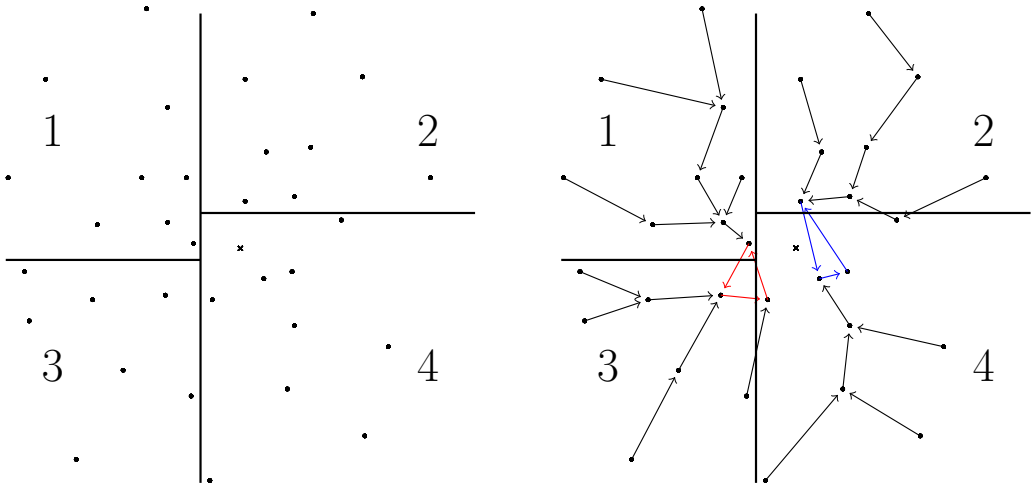
Figure 5.7: Final merge of unlinked haloes.

5.3 Parallelization

The basic algorithm is relatively straightforward to implement as a serial code. The parallel algorithm is much harder because of the following constraints and complications.

- All work must be done in parallel, and,
- All data must be fully distributed – there can be no "master" node with a complete list of all groups less memory on this node be exhausted.
- Groups will span processor boundaries, which implies that
- Individual chains can span processor boundaries, and,
- Terminating loops can even span processor boundaries.
- Results need to be identical regardless of the number of processors used, and thus need to be identical to the serial version.

The problem can be more clearly seen in figure 5.8. In figure 5.8a the particles have been broken into four domains corresponding to four hypothetical processors. In a real simulation then could be thousands or tens of thousands of domains. Figure 5.8b shows the results of the original group finding with chains, spurs, and terminating loops all crossing processor boundaries. In practice, many groups will be entirely contained on a single processor, but there will be a non-insignificant number that experience one or more of these characteristic cross-boundary cases.



(a) An example ORB decomposition of the particles.

(b) Illustration of how links and loops can cross processor boundaries.

Figure 5.8: Parallel Domains

5.3.1 Density/Gradient Calculation

The parallel implementation of grasshopper was added to pkdgrav2 which already had a working parallel "smooth" algorithm. The calculation of densities and density gradients only required the addition of new smoothing kernels.

5.3.2 Linking Particles in Chains

This process begins as with the serial version. A particle is chosen and a ball search is performed to find the N nearest neighbours. Some of these neighbours may be on remote processors, while some (or often all) will be local. The correct particle is chosen in the direction of the gradient, and if it is local then the algorithm continues identically as with the serial version.

If the particle is remote, then the link is update to point to the remote particle, and the algorithm continues with the next unlinked particle, as if it had found a terminating loop. This results in "groups" that are incomplete and may have:

- Particles linked to from one or more remote processors, or from several "groups" from the same remote processor.
- Be linked to a remote group.
- Be linked to another group on the same processor via particles on a remote processor (something we don't know about yet).

This can be seen in figure 5.9 where links that remain unresolved are indicated with dashed lines. This results in the following "groups":

1. Domain 1 has a single group. It points to a particle in domain 3.

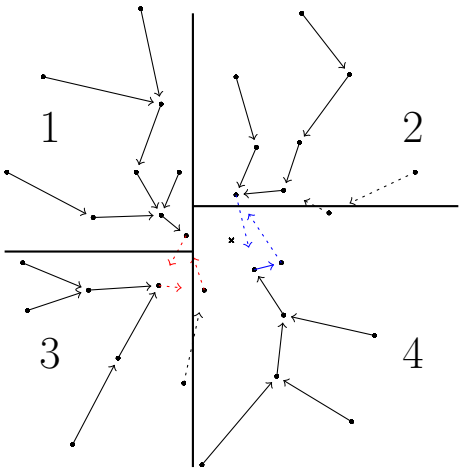


Figure 5.9: Results after particle linking phase. Dashed line indicates unresolved links.

2. Domain 2 has two groups, both of which point to different particles in domain 4.

3. Domain 3 has two groups, both of which point to a particle in domain 4 (in this case the same particle).

4. Domain 4 has only one group which points to a particle in domain 2.

It should be pointed out that at this point, the only remote information required is for the neighbour search. This is a fetch only type operation, and is reasonably fast — on the order of a gravity calculation. This leaves unresolved links, but this is addressed next.

5.3.3 Chain Resolution

At this point, all particles are members of a local group, and all local groups point to a single particle, either a remote particle, or an arbitrary particle in the terminating loop. To resolve these links we need to traverse the chain, but the process is **greatly** accelerated because complete chains are already precomputed on each node. The process is as follows for each unresolved group.

1. We fetch the remote particle we point to, and the group to which it belongs.

2. We fetch the remote group information. This group points to the next particle in the chain.

This processes is continued iteratively, traversing from particle to particles until either:

1. We terminate at a particle that has a group that is entirely local. In this case, our group becomes a member of this group (we are a remote subordinate).

2. We encounter a terminating loop that crosses processor boundaries. In this case, we chose one of the groups in the loop in such a way that all processors will chose the same group. For example, the lowest processor number can be selected always.

The detection of a loop that spans multiple processors[1] is achieved using Floyd's cycle finding algorithm, also called "Tortoise and Hare". The idea is that we traverse the links with two separate pointers. On each iteration, we move one a single link forward (the hare), and the other two links forward (the tortoise). If these pointers are ever identical, then we know that we have encountered a cycle, and that the current remote group is part of this cycle. We need only follow the links back around to the same point in order to identify the "master" group.

By now we can still have multiple local groups, that actually belong to the same group. They will all point to the same remote group (or the same one on our own node) so it is a simple matter to merge them together. Groups are now one of two different classes:

1. They are a local group (possibly with remote particles), or,
2. They point to a remote group which is a local group.

So at most, the level of indirection is one.

5.3.4 Chain Merging/Linking

The final phase to to join multiple loops that are close to each other, as in the serial algorithm. This is done by creating a tree with only particles that form terminating loops, and then looking for neighbours that are within a certain linking distance that are not part of the same chain. These chains are joined to form the final set of groups with results identical to the serial algorithm.

The choice of linking length is a free parameter, but in practice using several times the particles softening produces good results.

5.3.5 Unbinding

The unbinding process is much more complex than in the serial case, but fortunately most of the framework for calculating gravity is already in place as pkdgrav2 was designed for this. The situation is more complicated though. With pkdgrav2 , a single tree is built, and gravity is calculated for all active particles, based on force contributions from all particles.

In the case of grasshopper , the force calculation must be performed, but only taking into consideration particles within the same group. The most efficient way to calculate these forces is to create a separate tree for each group, and evaluate each separately. As we discovered earlier, and it is intuitively obvious that some of the groups will span processor boundaries. This was handled in the general pkdgrav2 case by creating a "top tree" containing cells that span multiple processors, eventually culminating in the "root" cell.

For grasshopper we take a slightly different approach. When a group exists on multiple processors, there is a designated "owner" processor. We send any root nodes from other processors to the master processor. These roots are then collected there and redistributed back as a complete list of tree roots. These tree roots are immediately added to the check

[1] It can be an arbitrary number in theory, but in practice will be only two or three.

list for each group, and they are walked as if the cells had already been opened. It would be theoretically possible in the worst case to need to send a root cell from each processor which would result in the storage going as the number of processors, but because grasshopper is a **sub**-halo finder, in practice the number of processors that a sub-halo spans is quite small.

Having calculated the gravitational potential energy, we now need to calculate the kinetic energy of the particles in order to determine the total binding energy. We know the velocity of all particles in the group, but in order to calculate the kinetic energy, we need to subtract the bulk velocity of the halo. To calculate the bulk velocity, we calculate the center of mass, and use the center of mass velocity as an estimate of the bulk velocity. For a completely bound group, the center of mass velocity **is** the bulk velocity, but this is not necessarily true for a group that contains unbound particles.

The serial version removes a single particle at a time until it has removed a certain number, then the kinetic energy of the particles is recalculated with the new center of mass velocity and the process repeated until there are no remaining unbound particles. Then the gravitational potential energy is recalculated with the new, small group and this is repeated until the process results in no additional particles being removed, or until the group has been completely evaporated.

This does not work very well in parallel for a couple of reasons. It is difficult to determine the particle with the highest binding energy when the group spans multiple domains in an iterative way without a severe performance hit or without using large amounts of memory. Finding the least bound particle either requires a synchronisation phase each time, which is extremely slow, or it requires that the list of least bound particles be sent to a single processor and ordered there.

The alternative is to apply a heuristic that approximates the solution while being much faster. We calculate the binding energy of the median unbound particle on each processor. These medians are exchanged and the one with the highest energy is chosen as the threshold. All particles with a binding energy higher than this are removed or unbound, and the centre of mass velocity and kinetic energies are recalculated. This process is done iteratively for each group until no unbound particles remain. This algorithm requires $O(\log_2 n)$ synchronisations, and scales as $O(n \log_2 n)$ where n is the number of particles that need to be unbound.

5.4 Parameterization

There are only two parameters that affect the algorithm: N_{smooth}, the number of neighbors over which to smooth, and τ, the linking length. The choice of N_{smooth} works well with 80 neighbours. If this parameter is set too low, then particle noise can result in the detection of false substructure. If set to high, then substructure will be missed.

The τ parameter is normally set to 3 or 4 times the particle softening length as discussed previously.

5.5 Friends of Friends

The underlying framework for group-finding was also used to provide a Friends-of-friends group finder. This group finder was used during the Euclid Flagship run on 2 trillion particles. An entire group-finding phase took 209 seconds compared to 237 seconds for a single gravity calculation and resulted in 7.8 billion groups.

Chapter 6

PKDGRAV3

submitted to Computational Astrophysics and Cosmology
Douglas Potter[1], Joachim Stadel[1], Romain Teyssier[1]
[1]Institute for Computational Science, University of Zurich, Winterthurerstr. 190, 8057 Zurich, Switzerland

We report on the successful completion of a 2 trillion particle cosmological simulation to z=0 run on the Piz Daint supercomputer (CSCS, Switzerland), using 4000+ GPU nodes for a little less than 80h of wall-clock time or 350,000 node hours. Using multiple benchmarks and performance measurements on the US Oak Ridge National Laboratory Titan supercomputer, we demonstrate that our code PKDGRAV3, delivers, to our knowledge, the fastest time-to-solution for large-scale cosmological N-body simulations. This was made possible by using the Fast Multipole Method in conjunction with individual and adaptive particle time steps, both deployed efficiently (and for the first time) on supercomputers with GPU-accelerated nodes. The very low memory footprint of PKDGRAV3 allowed us to run the first ever benchmark with 8 trillion particles on Titan, and to achieve perfect scaling up to 18000 nodes and a peak performance of 10 Pflops.

6.1 Overview of the Problem

The last decade has seen the advent of high precision cosmology, mostly because of the very accurate Cosmic Microwave Background (CMB) experiments WMAP [188] and Planck [1]. Cosmological parameters, such as the total matter content in the Universe or the Hubble constant are now constrained to within several percent. Although our best fit model, the so-called standard Lambda Cold Dark Matter (LCDM) model, very successfully explains these remarkable observations, it is still based on two mysterious, undetected and elusive components: dark matter and dark energy. The cosmological experiments of the next decade might shed light on this "dark sector" and possibly revolutionize modern physics. After a decade of CMB experiments, we expect large scale galaxy surveys, such as the ground based Large Synoptic Survey Telescope [119] (LSST), or the two satellite missions Euclid [115] (in Europe) and WFIRST [190] (in the US), to give new, stronger constraints on our standard cosmological model parameters, possibly below the percent level. Two techniques are considered to measure the clustering of matter as a function of time and scale: weak lensing (WL) and galaxy clustering (GC). Both techniques rely on very accurate

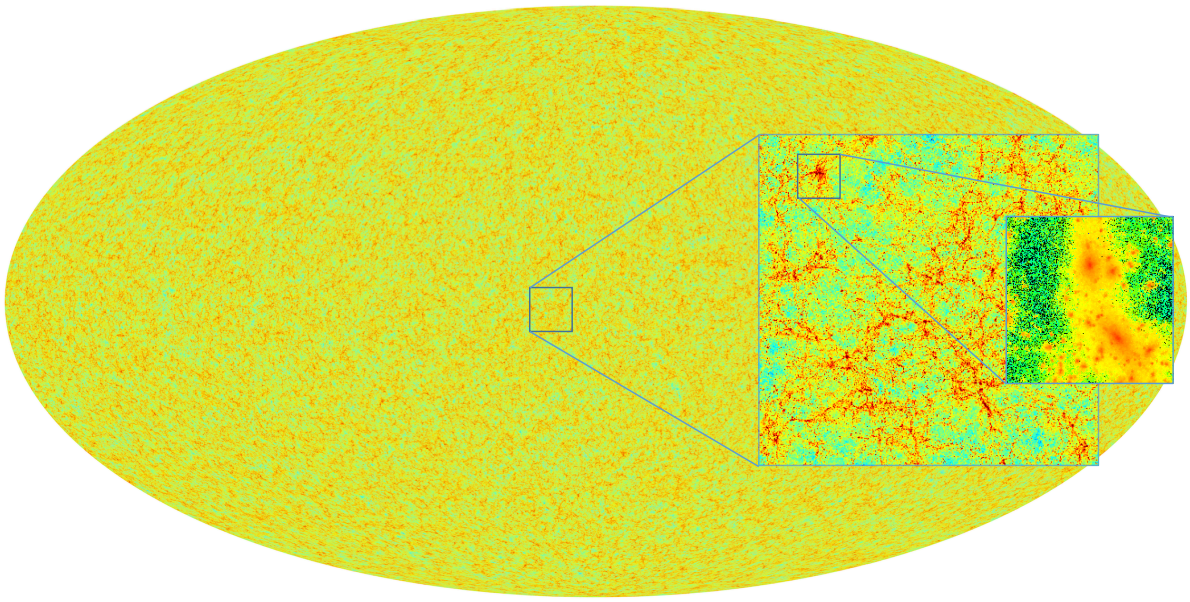


Figure 6.1: Simulated full-sky matter distribution from a 2 trillion particles simulation. The zoom-in quadrant shows the non-linear, filamentary structure of the universe on small scale.

theoretical predictions of the non-linear dynamics of the dark matter fluid in an expanding Universe. The more accurate these theoretical predictions are, the more efficient the future large scale surveys will be in solving the mysteries of the dark universe.

Because of the non-linear nature of gravity on these scales, our best theoretical predictions make use of N-body simulations: the dark matter fluid is sampled in phase space using as many macro-particles as possible, each one representing a large ensemble of true, microscopic dark matter particles, evolving without collision under the effect of their mutual gravitational attraction. We review in Section 6.2 the current state of the art in the development of high performance N-body codes. Motivated by future dark energy missions, our main goal is to reach an accuracy better than 1% in the power spectrum of the matter density field from linear scales (> 100 Mpc/h) down to strongly non-linear scales ($\simeq 1$ Mpc/h). For us to reach these extreme accuracy requirements, we face four different computational challenges: 1- high precision in the gravity calculation, 2- high accuracy in the time stepping, 3- reduce the statistical errors below 1%, which translates to a physical volume of $L \simeq 2$ Gpc/h, and 4- high enough mass resolution, that translates to a large number of particles (for a review see Ref. [169]). The last requirement pushes the limits of what can be achieved on current supercomputers: we need to model accurately dark matter haloes as small as one tenth of the Milky Way mass, which translates into a particle mass smaller than 10^9 $M_\odot$/h, and, for the adopted minimum box size, into a total particle count

of $N > 2$ trillion. In the context of future large galaxy surveys, we will need these extreme N-body simulations not just once, but for many different cosmological models, exploring alternative gravity models or galaxy formation scenarios. An additional requirement is a fast enough time-to-solution, so that N-body simulation can optimize and analyze cosmology experiments.

In this paper, we report on the successful evolution of a 2 trillion particles simulation of the LCDM model from $z = 49$ to $z = 0$ *in less than 80h of wall clock time* including on-the-fly analysis, performed on the the Swiss National Supercomputing Center Machine, Piz Daint, using 4000+ GPU-accelerated nodes. We also report on the first ever benchmark of a 8 trillion particles simulation of the same model, performed on Titan at Oak Ridge using 18000 GPU-accelerated nodes. Although our 2 trillion particles run represents the minimum requirements for future galaxy surveys, we establish the feasibility of even more extreme particle counts with our 8 trillion particle benchmark. Our tests demonstrate a significant reduction in the time-to-solution and put us in an ideal position to use these extreme N-body simulations for the preparation and the analysis of large galaxy surveys.

6.2 Current State of the Art

N-body simulations in astrophysics have been at the forefront of high performance computing, even before the first digital computer, with the galaxy collision experiment of Holmberg [94], based on moving light bulbs, and then the heroic 300-particle computer simulation of the Coma cluster performed by Peebles in 1969 [148]. Cosmological simulations have been particularly efficient at exploiting the best of each generation of supercomputers, adapting the algorithms to new architectures. In that respect, the number of simulated bodies (or particles) has increased dramatically, owing to the ever increasing performance of supercomputers, but also to the growing efficiency of the N-body solvers. Here, we report the first benchmark ever performed on 8 trillion (8×10^{12}) particles.

In the early 80's, gravity calculations quickly moved away from the accurate but slow $\mathcal{O}(N^2)$ direct interaction (where N stands here for the number of simulated particles) or Particle-Particle (PP) approach, to faster techniques, such as the Particle-Mesh (PM) algorithm [92], based on the Fast Fourier Transform (with $\mathcal{O}(N \ln N)$ efficiency) or the tree code [14] (also with $\mathcal{O}(N \ln N)$ scaling). Since the PM technique suffers from the limited resolution of the mesh, a hybrid version of PP and PM was later developed, leading to the P^{3}M technique, which is $\mathcal{O}(N \ln N)$ on large scale and $\mathcal{O}(N^2)$ on small scale [37]. The attitude of many generations of code developers since then was to take advantage of the shear performance of the best available computer at that time, but also to reduce drastically the time-to-solution by developing more complex but more efficient algorithms.

In that respect, cosmological simulations are particularly challenging, since they require a fixed simulation time of 13.7 Gyr, namely from the Big Bang until our present epoch. They also require, as explained in Section 6.1, the largest possible number of particles that can fit in the computer memory. This has led computational cosmologists to develop clever and innovative solutions to optimize the gravity solvers.

Warren and Salmon were among the first cosmologists to be recognized for their parallel tree code's performance, reaching 430 Gflops on ASCI Red [211, 134]. In 2012, The Millennium XXL simulation[8] was run with 0.3 trillion particles using a specialized version of

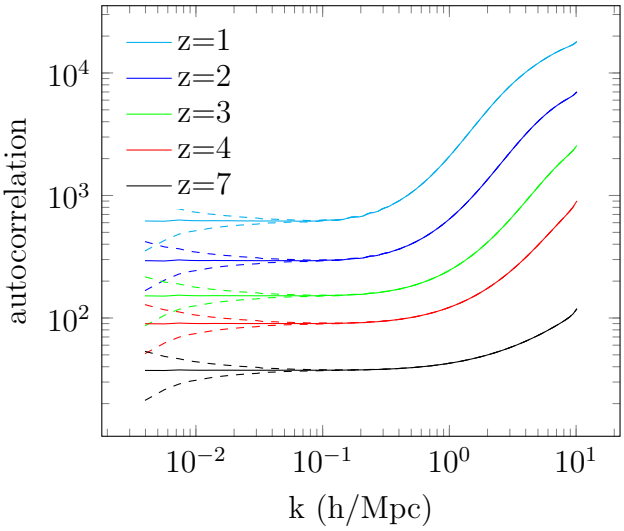


Figure 6.2: Auto-correlation functions of the density fluctuation in a cosmological box of 3 Gpc sampled with 2 trillion particles simulation at various epoch indicated by the redshift. Dashed lines indicate the statistical errors due to the finite volume of the simulation. The accuracy of these theoretical predictions is far below the percent level on almost all scales.

the GADGET-3 code, based on GADGET-2[192]. At about the same time, Ishiyama *et al.* also achieved 4.5 Pflops with a 1 trillion particle simulation run on the K computer [97] for a cosmological simulation using GreeM [96], another parallel tree code. Habib *et al.* [78] performed a 3.7×10^{12} particle benchmark on a BG/Q system in 2013, this time with a new generation PM+X[1] code called HACC. The HACC code was used in 2014 to produce the Q Continnum Simulation[88]; a full cosmological simulation of 0.55 trillion particles. In 2014 another 1 trillion particle simulation was run by Skillman *et al.* [182] using the 2HOT code [212]. More recently, Bedorf *et al.* [17] developed a tree code fully ported on GPUs, and delivered almost 25 Pflops on the Titan supercomputer. These recent achievements demonstrate that tree codes and P^{3}M codes, both scaling as $\mathcal{O}(N \ln N)$, can deliver significant performance on parallel, and more recently on GPU accelerated, hardware.

In parallel, however, new algorithms have been developed, both for particle and grid-based gravity solver, which in principle could reduce even more the time-to-solution for cosmological simulations. These are the Multigrid (MG) solver [30], which can replace the FFT advantageously, as it scales as $\mathcal{O}(N)$, and the Fast Multipole Method (FMM) [75, 48] which could deliver the same $\mathcal{O}(N)$ scaling for tree-based codes. While the former, implemented in the Adaptive Mesh Refinement code RAMSES [201], has been used recently in the 500 billion particles cosmological simulation DEUS [4], the latter, implemented in the PKDGRAV3 code, is the main subject of the present paper.

The $\mathcal{O}(N)$ scaling of FMM clearly offers the opportunity to go to higher particle counts, or to reduce significantly the time-to-solution for a fixed N. Since cosmological simulations are targeting the highest possible value for N, memory is also a strong limitation. The main innovations presented in this paper are 1- a highly performing version of the FMM algorithm, with a measured peak performance of 10 Pflops, and 2- an optimal use of the

[1]HACC can use a number of hybrid PM methods including P^3M or TreePM with or without GPU or other accelerators.

available memory, allowing us to reach 8 trillion particles on the 18000 nodes of the Titan supercomputer.

6.3 Algorithmic Improvements

6.3.1 Fast Multipole Method

As the "N" in N-body simulations has increased into the trillions, the asymptotic order of the algorithms to calculate the gravitational forces between the particles is central to having a fast time-to-solution. The $\mathcal{O}(N \ln N)$ gravity calculation of Barnes-Hut (BH) tree-codes, even highly optimized ones which achieve excellent peak performance, are problematic for cosmology simulations. FMM is now vastly superior to the BH for large N, even though it has somewhat lower peak floating point rate than measured by some recent BH codes (Bonzai[17], 2HOT[212]). An aspect of FMM for cosmology simulation is that unlike other codes (BH, P^3M, and tree-PM) the gravity calculation does not take longer as the simulation progresses from the early smooth state of the Universe toward the present day, highly clustered state of matter. This is because FMM *must*, by its scaling with N, be effectively "blind" to the depth of the tree structure, and hence to the degree of clustering present among the particles in the simulation. FMM and BH are very similar methods; both use particle-particle (PP) interactions for nearby particles and a multipole expansion of the mass within a more distant cell to approximate the force (PC-interactions). However, FMM also considers *cell-cell* (CC) interactions by approximating the potential "landscape" within a given cell (the sink cell) that is induced by a sufficiently distant multipole (the source cell). While any implementation which uses CC interactions in a sufficiently general way will scale as $\mathcal{O}(N)$ and thus qualifies as an FMM code, several key differences make the FMM as used in PKDGRAV3 highly efficient for very large N simulations.

FMM was originally implemented by Greengard [75] using a hierarchy of uniform meshes, but is in fact perfectly suited to implementation using a tree structure as in the BH method. Unlike most tree-codes, PKDGRAV3, uses a binary tree where parent cells are divided along the longest axis into two equal volumed child cells. Using a binary tree as opposed to an oct-tree provides a finer jump in accuracy when going from an expansion based on a parent cell to using the sum of expansions for the child cells. This leads to fewer terms being required to achieve the same force calculation accuracy at the expense of somewhat higher cost in making these decisions (tree walk phase). Another advantage is the simplicity of handling the non-cubical domains that result from *domain decomposition* which divides the simulation volume into sub-volumes which are local to each core. Since we use the traditional ORB (Orthogonal Recursive Bisection) decomposition to balance the number of particles in the domains, this forms the upper part of our global tree structure of which each node and core has a purely local subtree. In fact FMM naturally maximizes locality even within the memory hierarchy as it proceeds down the tree toward the leaf cells since the particles and cells are in a hierarchically sorted order after building the tree. Leaf cells of our tree contain up to b particles (we call this the *bucket size*), where the optimal value is around 16.

Central to the efficiency of a tree code, particularly one using GPU acceleration (see

below), is how we create lists of interactions (PP, PC, CC and CP[2]) which when evaluated give us the force on the particles. We walk the tree structure in node-left-right recursive order for sink cells (to which interactions apply) considering source cells that are collected on a checklist. Considering source cells for interactions is traditionally referred to as evaluating an *opening criterion*, but opening a cell (removing it from the checklist and adding its children to the end of the checklist) is only one possible outcome. A source cell on the checklist could also be put onto any of the four interaction lists depending on its distance from the sink cell, or it could remain on the checklist for further consideration by the *children of the sink cell* as we proceed deeper in the tree.[3] Evaluating the opening criterion is a purely arithmetic operation (using AVX/SSE intrinsics for performance and to avoid branches) resulting in a case value of 1 to 6 encoding the outcome for checklist elements. When done this way, these calculations are insignificant to the total computing cost ($\sim 2\%$). Tree walking begins with the sink cell being the root of the *local* tree of a processor while the checklist contains the *global* root cell of the entire simulation box as well as its 26 (and sometimes 124 depending on accuracy requirements) surrounding periodic replicas.

The actual opening criterion is critical in controlling the distributions of force errors, both in their magnitude and in their spatial correlations.[4] During tree build we calculate a bounding box for each cell and the distance, $b_{\max}$, from the center of mass of the cell (which is always the center of expansions in PKDGRAV3) to the most distant particle in the cell. Based on this we determine an *opening radius* for a cell, $RO = b_{\max}/\theta$, where θ is the traditional opening angle and the force accuracy controlling parameter in the code. If the distance between the source and sink (between centers of mass) are greater than $1.5RO_{\text{sink}} + RO_{\text{source}}$ *and* the bounding boxes are no closer than twice the *softening* (we use 1/50 times the mean inter-particle separation – for a review on the role of softening in N-body simulations see [53]), then this is a CC or CP interaction. Note, that there is a deliberate asymmetry here, the factor of 1.5, which controls the spatial correlations in the force errors. For a traditional BH code the force errors typically add up from all directions about a given particle and tend to be correlated spatially with the density of particles. For FMM on the other hand, there is almost no correlation with density (again a working FMM must be blind to tree depth), but we see the tree structure since the expansion of the potential within a sink cell is most accurate at the center of mass and degrades toward the edge of the cell. To reduce this spatial correlation below about 10% of the random errors we have made the acceptance of CC and CP interactions stricter by making sink opening radii larger by this factor. If leaf cells are opened their particles are added to the checklist with $RO_{\text{source}} = 0$ and can later become CP or PP interactions. If a source cell is reached with fewer than g particles (called the *group size*) we proceed no deeper in the tree resolving the remaining checklist into interaction lists, including now PP and PC as well. We have found that a group size of 64, or more generally four times the bucket size, seems to be close to optimal for PKDGRAV3.

Most tree-codes consider multipoles of up to only 2nd order (quadrupoles) which is most efficient for low accuracy force calculation, however for the needed force accuracy of

[2]Cell-particle interactions are the mirror image of particle-cell interactions; they are the expansions of the potential within the sink cell induced by a single source particle.

[3]It is rare for a cell to stay on the checklist for more than a few levels as it will end up on one of the interaction lists or be opened.

[4]Ideally we want spatially uncorrelated errors, but this is as impossible to attain as is having all force errors precisely at the desired truncation error.

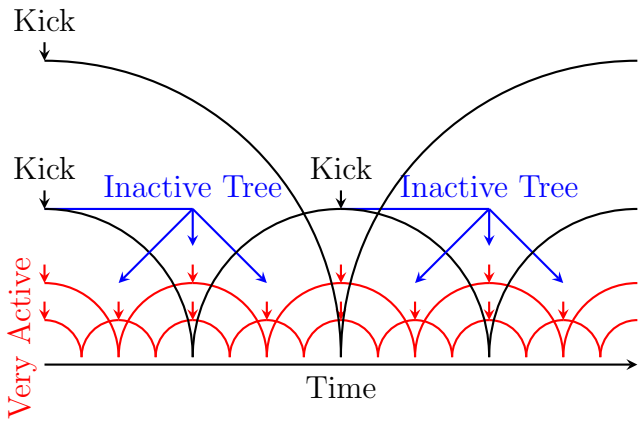


Figure 6.3: The kick-drift-kick multi-stepping "umbrella" diagram with the use of dual trees over a single base time-step. Each level of arcs represents one rung and domain decomposition is allowed to move particles between threads *only* at the apex of the black arcs. At these points a single tree is built to for all particles in the usual way. Next an inactive (or fixed) tree is built halfway through the black interval and used to calculate force contributions to the remaining red time-steps in a time symmetric way as shown in blue. The red, very active, subtree is all that is built on the shorter very active time-steps where both trees are walked to obtain the combined force.

better than 0.1% RMS, going to 4th order moments is more than twice as efficient [195, 48]. Not only does the flop/byte ratio increase with order, but also the ratio of FMA (fused multiply add) operations to regular multiply/add, and the number of those compared to the one required $1/\sqrt{|\mathbf{r}|^2}$ increases substantially. The local expansion of the potential about the sink's center of mass is actually done to 5th order, but we do not store this in the tree, since it is sufficient to keep it as a local variable accumulating the CC and CP interactions as we walk the tree. We use single precision in calculating interactions, but all components are accumulated in double precision so we can achieve force errors of around 10^{-5}%, well below what is needed for these simulations. To implement periodic boundary conditions, PKDGRAV3 uses a 5th order multipole approximation of the Ewald summation potential [195, 90, 106]. This requires virtually no data movement and is ideally suited to GPU acceleration, but these calculations must all be done in double precision. Our mixed precision approach serves both to reduce memory usage as well as maximizing the benefit from AVX/SSE as well as GPU floating point hardware.

6.3.2 Multiple Time Stepping with Dual Trees

Cosmological simulations span enormous ranges in density, from very underdense voids, to the centers of dark matter halos that can have densities of 5 orders of magnitude above the mean. This in turn implies that a huge range in dynamical time-scales exist within the simulation. Calculating gravity on all particles at every smallest time-step, while simple from the parallel computing stand-point is very wasteful if the the goal is fast time-to-solution for such simulations. PKDGRAV3 uses individual time-steps per particle, but restricted to being 2^{-l} times a certain *base time-step*, where l is the *rung* to which a particle belongs. All simulations presented here use 100 equal base time-steps in proper time to evolve the

simulated universes to the present, but many more time-steps are chosen for dynamically active areas of the simulation automatically. We use a hierarchical kick-drift-kick leap-frog scheme shown in figure 6.3, where the arrows indicate the force calculations that are applied to advance the velocities. Only the sink cells that contain particles belonging to rung l and higher need to be walked since kicks at higher rungs align in the diagram (we call these the *active* particles). We also need a time-step criterion to decide on which time scale a particle is evolving. The traditional one used in cosmology simulations is based on the particle's softening and the magnitude of its acceleration by $\Delta T_i = 0.2\sqrt{\epsilon/|\mathbf{a}_i|}$. It has been shown that the power spectrum [169] and mass functions of dark matter halos [159] converge using this time-stepping criterion. Given the distribution of particles in the rungs of a cosmological simulation, the potential speed-up that is theoretically possible is very large. However, due to the ever greater load imbalance, the decreasing flops/byte and the increase in the relative cost of overheads as the percentage of active particles decreases makes the speed-ups due to multi-stepping less dramatic, but still often a factor of 5x over much of the simulation. We discuss a novel method of reducing the most significant overhead, namely the tree build time, by building a second smaller tree only for very active particles.

With any multi-stepping code, there will be rungs with very little gravity work to do since only a small percentage of the particles are active. Nevertheless, the tree must still be built, walked, and the forces evaluated. The time needed for the force evaluations reaches a trivial stage while building a full tree still takes the same amount of time. As the number of tree builds scales as 2^l, the tree build cost quickly starts to dominate. We build a single second very active tree when the number of particles on a rung drop below a certain threshold (5% seems to be a good value)[5]. The inactive particles are drifted half-way along their trajectory and a fixed tree built as shown in figure 6.3. Subsequently, only an active tree is built until it is time to kick the fixed particles at which point they are drifted through the remaining half of their trajectory. It is very important to construct the second tree by traversing the fixed tree and using the same geometric structure. This assures that cells in the very active tree are approximately the same size as cells in the fixed tree in a given region of space (somewhat similar to the construction of *graded* trees in AMR codes). Not doing this sometimes results in an unreasonably high number of interacting particles.

6.3.3 GPU Acceleration

While other codes[16] have attempted to use the GPU for tree related operations, we made the deliberate decision to split the work between the CPU and GPU in a manner that compliments their strengths. Walking a tree is geometrically complex, exhibits branch divergence, and requires accessing tree nodes on remote processors. Conversely, evaluating interactions and multipoles is ideal work for the GPU. The GPU work consists of PP interactions, PC interaction and the periodic boundary condition evaluation (Ewald). PKDGRAV3 monitors the flop/byte ratio of interaction lists as they are generated and in the rare case that this falls below an optimal threshold then the work is instead issued directly to the CPU. This allows the GPU to concentrate on work packages that can keep utilization high resulting in a lower overall run-time. The operations are fully asynchronous allowing almost perfect overlap of compute and communication with the GPU.

[5]The dual trees are only constructed if there are at least two rungs below the fixed rung, otherwise there is no performance benefit.

6.3.4 Memory

With the use of FMM, multiple time-steps and GPU acceleration the major limiting factor for these simulations is the amount of available memory on each node. PKDGRAV3 has been developed to minimize memory usage per particle (see below) and allow the maximal use of the available memory for particles. This includes: 1- by-passing Linux file I/O and instead using direct I/O to have complete control of file buffering, 2- making memory balancing the primary goal of domain decomposition, 3- reducing the memory usage by the tree, 4- partitioning memory very carefully on a node and in most cases preallocating it. Careful consideration is also given to the memory usage of the many analysis tasks that are performed during the run including group finding, light cone generation as well as the storage required to generate the initial condition at the beginning of the simulation.[6] Minimizing the memory use per particle has the nice side benefit of increasing performance in the tree building and tree walking phases of the code that are strongly affected by the efficiency of transferring to and from memory.

Table 6.1: Memory requirements per particle.

Persistent	Ephemeral	Tree	Buffers
28 bytes	0-8 bytes	25 bytes	$\sim$ 5 bytes
Buffers are $\mathcal{O}$(125 MB) per thread. Here we assume 16 threads with $\mathbf{5 \times 10^8}$ particles on a 32 GB node.			

Storage for particles is divided into two regions; a "persistent" area containing properties that must persist between steps, and "ephemeral" storage used for certain algorithms, for example group finding, where the intermediate data can be forgotten when the calculation ends. In the persistent storage, we identified *position*, *velocity*, *group id*, and current *rung*. Velocities can be stored as single-precision float values without affecting the results. Positions are trickier. It is necessary to resolve well below the softening scale which in our case is one part in a million[7]. We would like to achieve a resolution of perhaps a hundredth of the softening length which would require of order 27 bits of precision, greater than that provided by single precision. We convert double precision float values between integer coordinates which provides 32 bits[8] of precision which is more than sufficient. We have checked that this simple particle compression scheme does not affect their trajectories in any significant way for cosmological N-body simulations. The ephemeral storage can vary between zero bytes (when no analysis is required), to 4 bytes if power spectra or group finding is needed up to 8 bytes for other algorithms. Future analysis may require more memory in which case the ephemeral area would increase. As a special case, it is possible to use part of the tree memory for algorithms when a tree is not required (when generating initial conditions for example). We also need a small amount of memory for explicit communication buffers as well as room for the tree (which tends to grow as structure forms). All told, the simulation can be run with approximately 62 bytes per particle as summarized in table 6.1. A simulation of 2 trillion particles can be easily run on Piz Daint (which has

[6]PKDGRAV3 uses the 2LPT method which requires 13 FFT operations and with some juggling can be done with 36 bytes per particle.

[7]Grid size of 1/20000 $\times$ softening scale of 1/50

[8]actually slightly less as the representable box must be slightly larger than the simulation volume

169 TB of memory) while an 8 trillion particle simulation can be run on Titan (which has 584 TB).

6.4 Performance Results

At the time this paper was written, Titan (Oak Ridge National Labs, USA) was the second fastest supercomputer in the world with a measured LINPACK performance of 17.59 Pflops and was used for most of the performance benchmarks reported here. It is a Cray XE7 system with 18'688 compute nodes and a Gemini 3-D Torus network. Piz Daint (Swiss National Supercomputing Center), a Cray XC30 with 5'272 compute nodes connected via the Aries Dragonfly (multilevel all-to-all) network is currently the 7th fastest computer in the world and is being used for the 2×10^{12} particle production run, upon which the benchmarks are based (the same mass resolution). The 282 node Cray XE6, Tödi (Swiss National Supercomputing Center), is useful for development and testing of large scale applications for Titan, being a much smaller instance of this system. The individual nodes of these three machines are similar, each having 32GB of main memory a single CPU as well as an nVidia K20X GPU accelerator. Titan and Tödi use the AMD Opteron models 6274 and 6272 with a clock speed of 2.2 and 2.1 GHz respectively while Piz Daint uses an Intel Xeon E5-2670 with a variable clock speed ranging from 2.6 GHz up to 3.3 GHz (3.0 GHz with all cores active). Titan has the largest total system memory of 584 TB which allows for a production simulation with PKDGRAV3 of 8×10^{12} particles with a time-to-solution of 67 hours. The detailed benchmark and scaling results presented below will establish that such a high resolution simulation is indeed possible within this projected time.

All of these machines have multiple CPU cores on each node, and the trend is for this number to increase. PKDGRAV3 employs a "hybrid" pthreads/MPI model with a single MPI thread per node, and threads on the same node exchange data using shared memory. While the dedicated MPI thread is only 25% utilized, not allowing it to participate in the gravity calculation has the effect of dramatically reducing message latency and increases overall performance.

6.4.1 Timing Measurements

In the following sections, timing information is collected through the use of timers in the code. The run-time is divided into four phases – load balancing, tree construction, force evaluations, and analysis. The first three phases are carefully timed and included in these results. The fourth, analysis, is not included as it can vary significantly depending on which analysis needs to be performed. If more sophisticated analysis "instruments" (by which we mean further software to perform on-the-fly analysis) were to be attached to PKDGRAV3 then the time would increase from the roughly 25% for our current production simulations.

We also use the high-resolution on-chip timers to measure sub-phases, in particular we are able to distinguish how much time is spend calculating forces, how much time is spent waiting for communication requests to complete, and how much time is wasted at the end of a step because of load imbalance. We discuss the later two only cursorily as they have a nearly insignificant effect on time-to-solution as shown in figure 6.4. The timings for analysis include the necessary I/O; indeed this can easily be seen in the figure where the

analysis time suddenly increases as the "particle light-cone" begins. Raw particle output is written to disk only when checkpointing which takes takes 30 minutes per checkpoint for the two-trillion particle simulation run on Piz Daint. This accounts for a roughly 5% cost increase depending on how frequently checkpoints are written. Initial conditions are also generated by PKDGRAV3 in memory at the start of the simulation, a procedure which takes approximately 5 minutes.

6.4.2 Simulation Accuracy

While it is possible to speed-up the simulations by relaxing the accuracy requirements, taking either fewer time-steps or increasing θ, thereby reducing the force accuracy, we emphasize here that we do not do this in any of the benchmarks. We run all benchmarks with the *same* run parameters that we are using for our 2×10^{12} particle production simulation which will serve as the first reference simulation for the Euclid mission. At very early times ($z > 20$), when the Universe is very homogeneous, the forces from opposing directions very nearly cancel and a tree code must use a stricter opening criterion in order to attain the same accuracy in the force. Additionally, small errors in the initial non-linear growth of these first structures amplify during the further evolution and can lead to errors greater than 1% in the power spectrum by the end of the simulation if the force accuracy and time-stepping is not conservative enough. We set $\theta = 0.40$ for $z > 20$ (to 1% age of the Universe), $\theta = 0.55$ for $20 > z > 2$ (to about 20% age of the Universe), and $\theta = 0.70$ for the remaining 80% of the evolution. We note that these quoted θ values apply for the 5^{th} order expansion used in PKDGRAV3 and result in much more accurate forces than in the traditional quadrupole based BH codes. These transitions in the force accuracy and cost per step can clearly be seen in figure 6.4.

The particle mass remained fixed at 10^9 solar masses for all benchmarks as previously mentioned. This is small enough to converge on the power spectrum to 1% and to resolve objects down to the needed scale to produce so called mock galaxy catalogues [66] for Euclid, weak lensing maps and statistics for galaxy clusters. It should be strongly emphasized that the smaller the mass scale that is simulated, the **harder** the simulation becomes, or comparing simulations of the same N, the one with the **smaller** box size is the more challenging. While PKDGRAV3 is independent of the degree of clustering in the force calculation, the peak densities within a simulation of smaller particle mass are higher and therefore the number of time-steps needed increases. We find that for PKDGRAV3 decreasing the box size by a factor of two while keeping the same number of particles results in an approximately 50% longer runtime.

In figure 6.4 we show the actual time spent in different tasks integrated over each base time-step for our completed 2×10^{12} particle production run on Piz Daint. Force calculation (in red) dominated the early time-steps, while later there is a near equal balance between it, tree building (green), domain decomposition (blue) and all of the on-the-fly analysis (magenta). The yellow/black contribution shows time spent waiting either because the work is not completely balanced (black), or because of communication delays (yellow).

It used to be the case that analysis was performed by post-processing the results, but with the ever increasing simulation sizes writing raw simulation output to the disk is no longer feasible, since this would vastly dominate the time-to-solution. The spike in the magenta analysis time at around step 20, for example, is a result of particle "light cone"

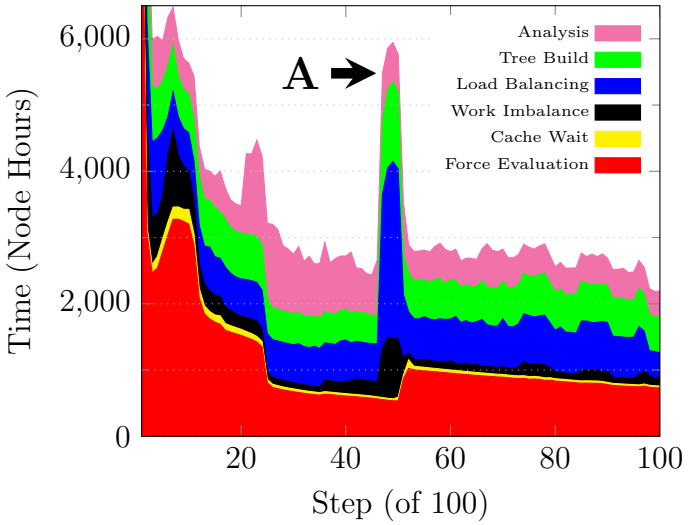


Figure 6.4: Distribution of run-time between various phases of the calculation. The red, yellow and black regions are force calculation, the blue region is for balancing the work, the green region is tree build and the magenta region is on-the-fly analysis. The feature indicated by **A** is described in Appendix 6.6.

analysis kicking in. Our friends-of-friends group finder, and the analysis on the resulting dark matter halos that are found by it, were also completely rewritten to be competitive with the other tasks (otherwise it would have been the dominating task at this scale). It is interesting to see that such analysis tasks must not be neglected when considered fast time-to-solution, since even when highly optimized, they contribute significantly to the total run time.

While tree building and domain decomposition times remain reasonably constant, gravity calculation changes for two reasons. As mentioned previously the force accuracy requirement changes (most notably at around step 24) when much of the mass is in viralized dark matter halos. The second reason is that the time-step also scales with the mean density of the Universe ($\Delta T \propto 1/\sqrt{\rho}$) which is decreasing very rapidly early on. This means that at the beginning of the simulation there are a lot of particles at very small time-step rungs which results in a heftier gravity calculation contribution. This never stops so the time per step will continue to decrease by a modest amount until the very end. We note again, that this is quite in contrast to what is observed for BH and P^3M codes. The onset of structure formation, which goes in the other direction to increase the number of time-steps, can be seen between steps 5 and 10 when the gravity time increases even though there has been no change in the force accuracy during this time. Structure formation stabilizes, in the sense that all density peaks have been established and most of the mass that can end up in dark matter halos is bound up in them [9]. Finally, the modest cost of tree building seen here is only possible when using the dual tree method described previously. Without this innovation the tree build contribution would be 3 times larger.

[9]Larger and larger structures continue to form but this does not affect the time-step hierarchy.

6.4.3 Multi-Stepping and Dual Tree Boost

Although there were 100 base steps, PKDGRAV3 uses a multi-stepping scheme where particles choose their own time-step rung based on the time-step criterion discussed previously. For the benchmark simulations this results in effectively $5000 \pm 10\%$ time steps. For rungs with very few particles, each step can take a fraction of a second. While the time for a full gravitational calculation can be in the range of minutes, the average time per step is of order 50 seconds, including tree build and domain decomposition (but not including on-the-fly analysis). For simulations of this type, multi-stepping results in an effective speed-up of between 4x and 5x when compared to taking single time-steps.

As discussed earlier, the tree building phase can begin to dominate when multi-stepping. A complete gravity takes of order two minutes, while constructing the tree takes more like 25 seconds. When multi-stepping, some of the gravity calculation takes less than a second while the tree building time does not vary. By constructing a second tree for the very active particles, the tree build time is reduced to one second for these critical sub-steps. The method results in an additional 26% decrease in the overall time-to-solution.

6.4.4 GPU Boost

PKDGRAV3 is already highly optimized for SIMD type instructions, such as SSE and AVX, and because of mixed-precision (float/double) code, the performance boost is already a factor of eight for some parts of the calculations. Because not all calculation are FLOP dominated, for example load balancing and tree construction, the effective speed-up is more like 3x. By using the GPU, the situation is dramatically improved. For the Tödi simulation shown in figure 6.7, a single force evaluation[10] that took 1138 seconds using only the CPU, takes 119.5 seconds when using the GPU – a speed-up of 9.5x. A complete step, including all phases (gravity, tree construction and load balancing), takes 1629 seconds with the GPU compared to 6507 with the CPU only, resulting in a 4.0x improvement in the time-to-solution.

Part of the GPU work scheduling involves shunting work to the CPU when appropriate. If the number of particles is too small (1 or 2), then the CPU will do the work. If the GPU is too busy, detected when too many work packages are scheduled on the GPU but not yet complete, then pieces of the interaction list that do not evenly align with a WARP[11] are done by the CPU instead. While it is possible to push more work to the GPU, and thus increasing the total FLOP rate, this comes at the expense of an increased time-to-solution.

6.4.5 Scaling

To perform the very largest simulations, it must be demonstrated that PKDGRAV3 can efficiently scale up to the task. Weak scaling was measured by starting with a 1000^3 simulation (10^9 particles) and running it on two nodes to measure the gravity calculation

[10] At late time when gravity calculations no longer dominate the run-time; speed-up at earlier times is higher.

[11] If the interaction list has 655 elements for example, then 640 would be calculated by the GPU, and 15 by the CPU.

times. The simulation was then scaled upward by scaling the total number of particles and the total number of nodes by the same factor. The simulations run are outlined in table 6.2. Here we see that the total run time remains constant as the simulation size is increased, which is expected for an $\mathcal{O}(N)$ method which has low parallel overheads and good load balance. We include a direct comparison with the HACC[78, 79] and 2HOT[182] codes. The weak scaling runs for PKDGRAV3 were all performed with 4.7×10^8 particles per node, the HACC benchmarks with 0.32×10^8 particles per node, and the 2HOT simulation with 0.81×10^8 particles per node. As the weak scaling of these codes is essentially perfect, the total run-time does not change when using the same number of particles per node. This is the most relevant scaling for these types of cosmological simulations as it is typical to be memory limited due to the desire for high resolution as well as large volume. For the same simulation size, 1.0×10^{12} particles, the results from HACC, 2HOT and PKDGRAV3 are similar with a science rate (millions of particles per second per node) of 1.7 for HACC[12], 1.2 for 2HOT[13], and 3.8 for PKDGRAV3. As the HACC and 2HOT benchmarks are not particularly current we would expect that today improved results could be presented by these authors. When the total number of particles per node was kept fixed at 4.7×10^8 as was the case for the weak scaling tests, an entire simulation would run to completion in 67 hours **regardless of size**.

Table 6.2: Weak Scaling Performance on Titan with 4.7×10^8 particles per node. The science rate remains constant.

Nodes	N_p	Mpc	Time	Science Rate[14]
2	1.0×10^9	250	124.9	4.00×10^6
17	8.0×10^9	500	117.4	4.02×10^6
136	6.4×10^{10}	1000	117.9	3.98×10^6
266	1.3×10^{11}	1250	125.1	3.76×10^6
2125	1.0×10^{12}	2500	124.0	3.79×10^6
7172	3.4×10^{12}	3750	123.2	3.82×10^6
11390	5.4×10^{12}	4375	126.6	3.72×10^6
18000	8.0×10^{12}	5000	120.1	3.70×10^6

To measure strong scaling, we start with a series of simulations with 1000^3, 2000^3 and 3000^3 particles (10^9, $8{\times}10^9$ and $2.7{\times}10^{10}$) and run them on the smallest number nodes where they will fit (so 4.7×10^8 particles per node). The number of nodes is then incrementally increased. As shown in figure 6.5, PKDGRAV3 shows excellent strong scaling up to a factor of several hundred. This allows us to reduce the wall-clock time of simulations by up to a factor of a hundred or more by simply increasing the number of nodes. Recall that when using the most particles possible per node and hence the maximum wall clock time, a simulation will take approximately 67 hours. Using 10 times as many nodes results in only a 25% penalty meaning a simulation would take less than 10 hours. Using 100 times

[12]Private communication
[13]Table 1 of [182]
[14]in particles per second per node

as many nodes carries a 70% penalty, meaning a simulation would take slightly longer than an hour.

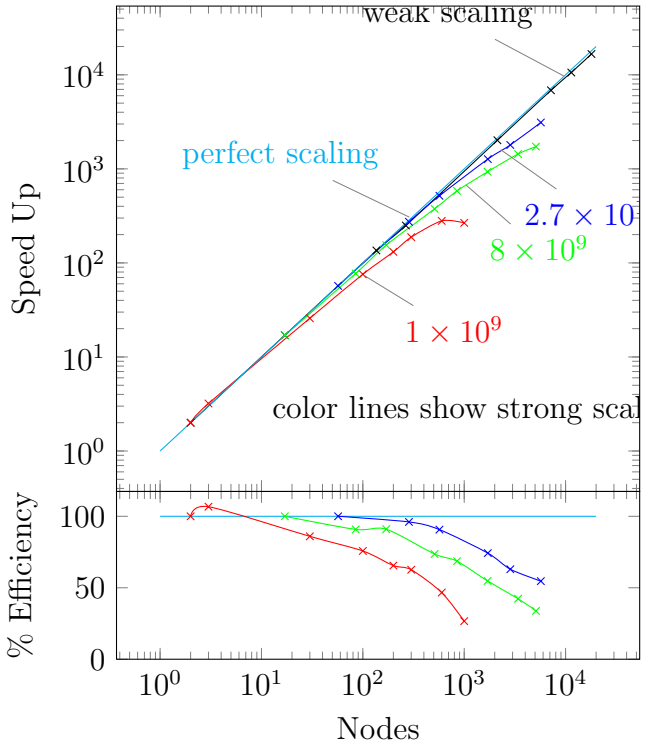


Figure 6.5: Weak and Strong Scaling. Perfect $\mathcal{O}(N)$ scaling should follow the slope of the "perfect scaling" line. PKDGRAV3 exhibits perfect weak scaling and excellent strong scaling out to 300 times the number of nodes. This translates into node memory usage starting at 30 GB and scaling to 0.1 GB.

6.4.6 Raw Performance

With PKDGRAV3, a great deal of effort has gone into algorithmic improvements to try to avoid, wherever possible, doing unnecessary work. This has the effect of greatly complicating the data structures making it more difficult to achieve high raw flop counts. Nevertheless, for a code to achieve high performance, the raw performance must be at least competitive.

To determine the number of floating point operations used, the AVX version of the code was examined to determine how many floating point instructions were required for each phase of the calculations. Most operations, including addition, subtraction and multiplication count as a single flop. The reciprocal square root is scored as seven flops while a division is scored as 35 flops. The totals for each phase are shown in table 6.3. In addition, floating point operations were divided into single and double precision, and totaled separately for the CPU and GPU.

In table 6.4 we show the peak performance achieved for various simulation sizes where the number of particles is optimized to fill a node. We also show the wall-clock time required to calculate the forces for a single particle.

Table 6.3: flop counts by phase

Phase	$+-\times$	$\sqrt{}$	$\div$	FLOPs
Particle/Particle	46	1		53
Particle/Cell	208	1		215
Cell/Particle	206	1		213
Cell/Cell	472	1		479
Ewald iteration	433	1	2	510
Opening criteria	97			97

Table 6.4: Performance on Titan. Total measured TFlops as well as the wall-clock time to calculate the forces for a single particle.

Nodes	N_p	Mpc	TFlops	Time / Particle
2	1.0×10^9	250	1.2	125 ns
17	8.0×10^9	500	10.3	14.7 ns
136	6.4×10^{10}	1000	82.2	1.84 ns
266	1.3×10^{11}	1250	152.5	1.00 ns
2125	1.0×10^{12}	2500	1230.3	0.124 ns
7172	3.4×10^{12}	3750	4130.9	0.0365 ns
11390	5.4×10^{12}	4375	6339.2	0.0236 ns
18000	8.0×10^{12}	5000	10096.2	0.0150 ns

While PKDGRAV3 does use mixed precision float code, the measured 10 Pflops compares quite well with the 17.59 measured LINPACK performance.

6.4.7 Time to Solution

To measure time-to-solution, we start by running a complete simulation at a lower resolution. Because of the physical processes involved, the timings for each step can be roughly broken into three distinct phases corresponding to different integration accuracy domains. In figure 6.6 we show the timings for gravity calculations in total node hours during each of the 100 main steps. As PKDGRAV3 is an $\mathcal{O}(N)$ code, these timings are then scaled linearly by the problem size to estimate how long the force calculations will take. The estimates are verified by running the force calculation at sampled points, and comparing them to the estimates.

This can be seen in figure 6.6. The hollow circles represent the measured timing for a force calculation on all particles throughout a simulation of $2500^3(1.5 \times 10^{10})$ particles. As is clearly apparent in the figure, the time required perform the gravity calculation is extremely stable. The three different "steps" correspond to the accuracy requirements (high redshift requires increased accuracy). The timings are given in node minutes (wall clock time multiplied by the number of nodes).

The dashed lines show predictions for the force evaluations at increasing resolutions made by scaling the low resolution simulation by the problem size. Measurements were

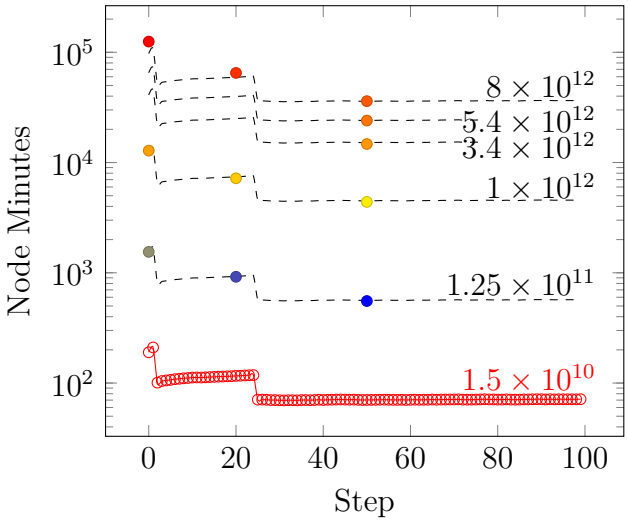


Figure 6.6: Gravity time per step. Circles are measurements while dashed lines are predictions.

then taken a several points at each resolution shown by the solid circles. The prediction and measurements agree perfectly.

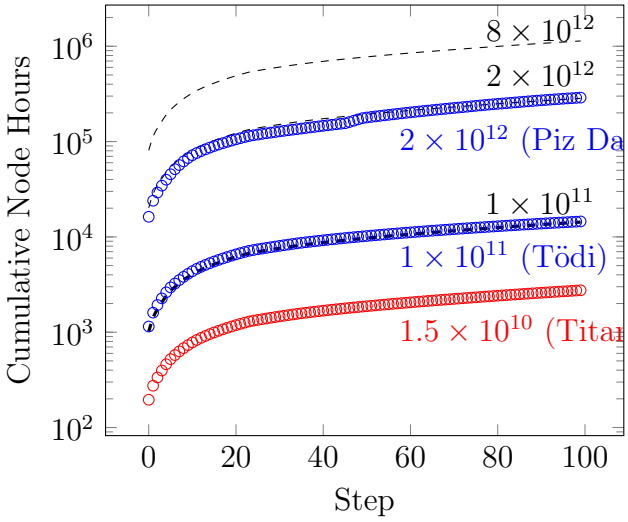


Figure 6.7: Total runtime in node hours (wall clock × number of nodes). The red circles are measurements of a simulation run on Titan with 320 nodes. The dashed lines are predictions based on both weak and strong scaling. The much larger Tödi simulation was run on 214 nodes while the even larger Daint run used 4900, and later 4000 nodes. All measurements show excellent agreement with the predictions.

In figure 6.7, the cumulative node hours for the reference simulation is plotted. In order to complete the simulation quickly, it was run on 320 nodes, even though it could have fit in as few at 32. We make predictions for how long simulations of various sizes, namely 10^{11}, 2×10^{12} and 8×10^{12} would take based on the weak scaling. As this is now in the strong scaling regime we further correct the prediction by assuming that it could be run 24% faster (recall that the penalty for strong scaling by a factor of 10 is 24%). These predictions are shown as dashed black lines.

A complete simulation of 10^{11} particles was run on Tödi using 214 nodes which corresponds to full memory usage of 4.7×10^8 particles per node. The measured performance of the simulation shows perfect agreement with the estimate. A further simulation was run on Piz Daint using 2×10^{12} (2 trillion) particles. Due to the slight differences in architecture, the simulation actually beats the prediction by a modest amount. We expect this is due to the slightly better AVX performance on the CPU, and perhaps to a lesser degree the network.

The end result is that we have high confidence that an 8 trillion particle simulation is possible on Titan using $18,600$ nodes, and it will take of order 67 hours with some additional time for on-the-fly analysis which would vary depending on the exact analysis done. This also means that a 1 trillion particle simulation run on Titan using all nodes could be completed in under 10 hours.

6.5 Implications

In order to achieve the results presented here, significant *refactoring* of the code was required. Tracking the progress in N-body simulations over time, a performance doubling time of roughly 1 year is observed. This rate, which exceeds Moore's Law can only continue if further efforts are made to refactor algorithms for new computing hardware. These gains can also be pushed forward by *co-design*, where computing hardware and algorithmic developments are considered as a single design process.

The new time-to-solution of these simulations is a game changer as far as the way theory is used in cosmological measurements. For the first time simulations will not only be used to help understand effects or to make some predictions, but will be needed to extract fundamental physical parameters from future survey data. They must become part of the data analysis pipelines.

Another implication for the future is that time-to-solution will continue to decrease as greater computational speed will out-strip any possible increase in memory size. Our memory footprint is about as low as it is possible to go per particle, so that the time-to-solution for these simulation can only decrease from this point on. We expect to run such simulations within 8 hours or less within the decade. This also means that raw data will never be stored and post-processed. Instead data analysis "instruments" will be attached to the code and the simulations will be rerun, perhaps several times with different "instrumentation". This is starting to happen and is a true paradigm shift in the field of simulations.

Acknowledgments

We are indebted to the folks at the Swiss National Supercomputer Center, specifically Thomas Schulthess, Maria Grazia Giuffreda and Claudio Gheller for the support and advice, and for resurrecting Tödi. We would also like to thank Jack Wells for arranging access to Titan. Parts of this work were supported by a grant from the Swiss National Supercomputing Center (CSCS) under project ID S592. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is

supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

6.6 Computational Challenges

During Grand Challenge simulations such as this one, there are inevitably problems encountered, and such was the case here. In figure 6.4, the time per step suddenly increases at step 46 as indicated by the arrow labelled **A**. This was caused by one of the nodes performing in a substandard way which resulted in the entire simulation to take twice as long, as the other nodes were waiting for this node to complete its share of the work. The exact cause of this problem is not known, and will never be known, but it was very likely a rogue process that was left running on the node that stole processing cycles. This problem disappeared when the simulation was restarted without this node.

The second problem occurred shortly thereafter, around step 50, and was a result of the increase in efficiency as the simulation progressed. In figure 6.4 we see that the gravity calculation time drops dramatically between step 0 and step 20 as structure forms and the effect of the initial condition grid is no longer relevant allowing the force accuracy to be relaxed. At some point, the amount of work being shipped to the GPU reaches a threshold that triggers a not yet understood problem with the GPU device. When this threshold is reached, the GPU will, very rarely, accept work but never complete it. By sending work in a more controlled fashion, this problem is eliminated or vastly reduced allowing the simulation to run to solution, but with slightly decreased performance. The cause of this is still under investigation.

Although these two problems seem dramatic, they had very little impact on the total run-time as can be seen in figure 6.7. The simulation was on track to slightly beat the estimate, but the two problems conspired to slightly increase the total run-time causing it to take almost exactly the amount of time predicted.

Chapter 7

GHALO

MNRAS: Letters, Volume 398, Issue 1, pp. L21-L25.

J. Stadel[1], D. Potter[1], B. Moore[1], J. Diemand[2], P. Madau[2], M. Zemp[2], M. Kuhlen[3] & V. Quilis[4]

[1]Institute for Theoretical Physics, University of Zurich, Winterthurerstr. 190, 8057 Zurich, Switzerland

[2]University of California, Department of Astronomy and Astrophysics, 1156 High Street, Santa Cruz CA 95064, USA

[3]Institute for Advanced Study, Einstein Drive, Princeton, NJ 08540, USA

[4]Departament d'Astronomia i Astrofísica, Universitat de València, Dr. Moliner 50, 46100 Burjassot (València), Spain

We perform a series of simulations of a Galactic mass dark matter halo at different resolutions, our largest uses over three billion particles and has a mass resolution of $1000 M_{\odot}$. We quantify the structural properties of the inner dark matter distribution and study how they depend on numerical resolution. We can measure the density profile to a distance of 120 pc (0.05% of $R_{\rm vir}$) where the logarithmic slope is -0.8 and -1.4 at (0.5% of $R_{\rm vir}$). We propose a new two parameter fitting function that has a linearly varying logarithmic density gradient which fits the GHALO and VL2 density profiles extremely well. Convergence in the density profile and the halo shape scales as $N^{-1/3}$, but the shape converges at a radius three times larger at which point the halo becomes more spherical due to numerical resolution. The six dimensional phase-space profile is dominated by the presence of the substructures and does not follow a power law, except in the smooth under-resolved inner few kpc.

7.1 Introduction

Over twenty five years ago the theoretical framework for the evolution of a cold dark matter (CDM) dominated universe was established [146]. The hierarchical and violent growth of structure in this model begins at a scale of $10^{-6} M_{\odot}$ [57] until the most massive clusters of galaxies form that are many orders of magnitude more massive. The assumption that the dark matter is cold remains to be verified, yet numerical simulations that follow the hierarchical formation of CDM haloes have given several fundamental and robust predictions for the structural and substructure properties of the dark matter distribution within virialised

haloes [60, 140, 11, 32, 68]. These results are widely used to compare with observational data and to assist comparisons with analytic models.

The first CDM halo simulated with enough resolution to resolve substructure used 10^6 particles [138], resolving the density profile to about one percent of the virial radius [69, 136, 153, 141, 56]. Whilst such simulations find numerous substructures in the outer halo, they find few or none within the inner 20% of $R_{\rm vir}$ and no obvious structure in phase-space in the central halo regions [137]. Advances in algorithms and supercomputing power have recently allowed us to increase this resolution by over two orders of magnitude with the Via Lactea II (VL2) simulation [59].

There are several reasons why we wish to carry out further studies at a higher resolution: (i) There are many old and forthcoming observational tests that constrain the structure of dark matter haloes on scales well within $0.001R_{\rm vir}$. These include high resolution rotation curve data and the kinematics of stars at the centres of dwarf galaxies. Future proper motions of these inner stars with GAIA or SIM will provide even tighter constraints. The close binary nuclei in galaxies such as VCC128 constrains the dynamics on even smaller scales [71]. (ii) As large surveys have pushed the surface brightness limits and detection efficiencies, many extremely faint satellite galaxies have been found orbiting the Milky Way. The completeness of current surveys is debated, and it has been argued that many hundreds of additional systems may be found in the coming years [204]. Simulations that can resolve and follow the survival of substructure within 10% of $R_{\rm vir}$ are necessary to compare with these data. (iii) Dark matter detection, either directly on Earth or indirectly via detection of annihilation relics, is the ultimate way to determine its nature. These experiments rely on accurate predictions for the phase-space structure of dark matter at the position of the Earth's orbit and the abundance and inner properties of substructure throughout the Galactic halo. (iv) Understanding the equilibrium structure resulting from violent relaxation is the ultimate challenge for galactic dynamicists. There is no compelling theory that can explain universal density and phase-space density profiles [199], or correlations such as between the local density profile and the anisotropy parameter [82].

Given this motivation, we have carried out a sequence of simulations of a single Galactic mass dark matter halo, which at our highest resolution contains over a billion particles within its virial radius. In this letter we report on its inner structure and convergence properties.

7.2 The Simulations

Our initial conditions are based upon the WMAP3+SDSS [189, 114] cosmological model with $\sigma_8 = 0.742, \Omega_M = 0.237, \Omega_\Lambda = 0.763, h = 0.735, n = 0.951$. The galaxy sized, $10^{12} M_\odot$, $R_{\rm vir} = 240$ kpc, halo was selected from a cosmological cube of 40 Mpc on a side. This simulation had 488^3 particles (simulation GHALO$_5$) in which three further nested spatial refinements by a factor of 3 (GHALO$_{4,3,2}$) were placed such that the Lagrangian region of about $3R_{\rm vir}$ of the halo at $z = 0$ was covered by 2.1×10^9 high resolution particles in the initial condition. The final effective resolution of GHALO$_2$ is 13176^3 resulting in a particle mass of 1000 $M_\odot$ and a total of 3.1×10^9 particles and 1.3×10^9 particles within $R_{200} = 347$ kpc. This allows us to capture all substructures out to more than $2R_{200}$ at the highest resolution. A further refinement GHALO$_1$ (in progress) will resolve the phase-space

Figure 7.1: The density of dark matter within the inner 200 kpc of GHALO$_2$. There are about 100,000 subhaloes that orbit within the virial radius. Each bright spot in this image is an individual, bound, dark matter subhalo made up of many thousands of particles (there are far more particles than pixels here).

structure at the position of the sun more sharply for future recoil dark matter detection experiments.

Creating these initial conditions was a significant challenge and we had to parallelize the GRAFIC1 *and* GRAFIC2 codes of [23] whereby the GRAFIC2 code was completely rewritten in C and MPI, and checked for near machine precision agreement with the original GRAFIC2. The new parallel GRAFIC1&2 codes can be obtained from the authors. Generation of the initial condition took 10 hours on 500 CPUs. We found that the original GRAFIC2 code had a bug in which the power spectrum used for the refinements was effectively that of the baryonic component. Although this has affected many previous simulations (not GHALO, nor VL2), tests show that the conclusions of these studies are not compromised.

The GHALO_2 simulation was run at the Barcelona Supercomputer Center on 1000 CPUs of Marenostrum using a total of 2 million CPU hours. Several significant improvements to the gravity code PKDGRAV2 made this calculation possible including much better parallel computing efficiency and SIMD vector processing. PKDGRAV2 uses a fast multipole method (FMM) similar to Dehnen [49, 48] but using a 5th-order reduced expansion for faster and more accurate force calculation in parallel, and a multipole based Ewald summation technique for periodic boundary conditions [195]. It uses adaptive individual time-steps for particles based on a new estimator of the local dynamical time [218]. The opening angle in the gravity tree and the accuracy parameter in the dynamical time-stepping is $\Theta = 0.55$ and $\eta = 0.03$ before $z = 2$, and then increased to 0.7 and 0.06 respectively. We make several comparisons to the VL2 simulation which was also run with the FMM version of PKDGRAV2, but whose initial conditions were selected and generated independently using somewhat different methods. The VL2 halo has a mass of $2 \times 10^{12} M_\odot$ and used a particle mass of 4000 $M_\odot$. The spline softening lengths for GHALO_2, VL2, $\mathrm{GHALO}_{3,4,5}$ are 61, 40, 182, 546, and 1639 pc, respectively (for GHALO these are set to 1/50 of the mean inter-particle separation).

7.3 The inner halo structure

7.3.1 The dark matter density profile

We apply a logarithmic binning to determine the radial density profile for the various simulations which are shown in Figure 2. The convergence radius of the density profile for the lower resolution realizations ($\mathrm{GHALO}_{3,4,5}$) can be clearly seen and are shown by the tick marks. These scale roughly as expected with $r_{\rm conv} \propto N^{-1/3}$, and we extrapolate this to conclude that the convergence radius of GHALO_2 is around 120 pc. The inner slope of GHALO_2 is -0.8 at 120 pc = 0.05% of $R_{\rm vir}$ and -1.4 at 2 kpc where the first subhalos become visible. Also shown is the power-law slope as a function of $\log(r)$, which exhibits a similar linear functional form for both haloes with no rescaling. Based on this observation we propose a new functional form for the fitting function of the density profile,

$$\rho(r) = \rho_0 e^{-\lambda(\ln(1+r/R_\lambda))^2} \tag{7.1}$$

which we term the S&M-profile (Stadel & Moore in preparation). It is linear in this plot down to a scale R_λ beyond which it approaches the central maximum density ρ_0 as $r \to 0$.

Fitting Function	Hernquist (α,β,γ) (ρ_s, R_s)	$\Delta^2(\times 10^{-4})$ GH$_2$	$\Delta^2(\times 10^{-4})$ VL2	$\rho\,[10^6 M_\odot kpc^{-3}]$ GH$_2$	$\rho\,[10^6 M_\odot kpc^{-3}]$ VL2	$R\,[kpc]$ GH$_2$	$R\,[kpc]$ VL2	3rd parameter GH$_2$	3rd parameter VL2
NFW	(1,3,1)	3.5	6.6	2.32	4.24	14.1	13.9	—	
Dehnen-McLaughlin	$(\frac{4}{9}, \frac{31}{9}, \frac{7}{9})$	1.6	0.70	0.273	0.591	42.6	36.7	—	
S&M-profile (ρ_0, R_λ)	—	**0.93**	**0.41**	5050	11000	2.20	1.88	—	
Generalized NFW	(1,3,γ)	3.0	2.7	1.78	1.87	16.2	20.9	1.04	1.13
Dehnen-McLaughlin	$(\frac{(4-2\beta_0)}{9}, \frac{(31-2\beta_0)}{9}, \frac{(7+10\beta_0)}{9})$	1.3	0.68	0.466	0.522	32.0	39.1	-0.0531	0.0129
Prugniel-Simien (ρ', R_e, α)	—	1.5	0.94	14.0	19.5	59.6	92.4	0.376	0.328
Einasto $(\rho_{-2}, R_{-2}, \alpha)$	—	1.0	0.45	0.685	0.991	26.8	28.9	0.155	0.142
S&M-profile $(\rho_0, R_\lambda, \lambda)$	—	**0.92**	**0.41**	4710	11200	2.47	1.82	**0.102**	**0.100**

Table 7.1: Fitting parameters and Δ^2 for each of the 2 and 3-parameter models for both GHALO$_2$ and VL2 simulations. Here $\Delta^2 = \sum_i^m (\ln(\rho_i) - \ln(\rho_{\rm MODEL}(r_i)))^2/(m-3)$ where ρ_i are the density values in logarithmically spaced radial bins at r_i. We fit from the resolved radius to 15% of $R_{\rm vir}$ at which point substructure begins to cause significant fluctuations in the profile. Consistent with [59] we obtain a generalized NFW with $(\rho_s, R_s, \gamma) =$ (1.05,28.0,1.23) (units as above) for VL2 by fitting from 360 pc to $R_{\rm vir}$, with the best fit profile being Prugniel-Simien over this range, $(\rho', R_e, \alpha) = (18.3, 113, 0.308)$.

We also note that if one makes a plot of $d\ln\rho/d\ln(1+r/R_\lambda)$ vs. $\ln(1+r/R_\lambda)$, then this profile forms an exact straight line with slope -2λ.

Table 1 lists the best fitting parameters for several functions: the S&M-profile, the restricted Hernquist (α,β,γ) profiles [89, 219], the Einasto profile [62, 141]

$$\rho(r) = \rho_{-2}\exp(-\tfrac{2}{\alpha}[(r/R_{-2})^\alpha - 1]), \tag{7.2}$$

and the Prugniel and Simien [156] profile

$$\rho(r) = \rho'(r/R_e)^{-p_\alpha}\exp(-b_\alpha(r/R_{-2})^\alpha), \tag{7.3}$$

where $p_\alpha = 1 - 0.6097\alpha + 0.05463\alpha^2$ and $b_\alpha = \frac{2}{\alpha} - \frac{1}{3} + 0.009876\alpha$ (for $\alpha < 2$, see Merritt et al. [133]) such that when projected one obtains a Sérsic profile [177, 176].

The residuals shown in Figure 2 show that the S&M-profile provides a slightly better fit than all the models for the inner, more consistent, part of the profile. Furthermore, it is the only 3-parameter model where the 3rd parameter has a consistent value for the two different simulations. For this reason we also list this model as a possible 2-parameter model, fixing $\lambda = 0.1$. The Einasto profile also provides an excellent fit to the density profiles of the two simulations.

7.3.2 Convergence of Halo Shape

The convergence of the shape parameters (see also Allgood et al. [6]) for GHALO in Figure 3 show that it is highly prolate over all resolved regions with b/a = c/a $\approx$ 0.5. At the halo centres the shape diverges quickly to a more spherical configuration. This is likely due to the orbital distribution being modified by the effects of resolution and softening. In this region the velocity distribution function is also strongly affected.

We estimate the convergence in the shape to be achieved at 0.3, 0.6, 2, 15 kpc for GHALO$_{2,3,4,5}$ respectively, a radius that is about 3 times the inferred convergence radius of the density profile but also scaling as $N^{-1/3}$. The fact that the variation in shape has little impact on the density profile can be understood by comparing the density profile taken in a 15 degree cone about the major, a, axis and the minor, c, axis [101]. The Δ^2 for the fits to the various density profiles remains roughly consistent between the two axial density

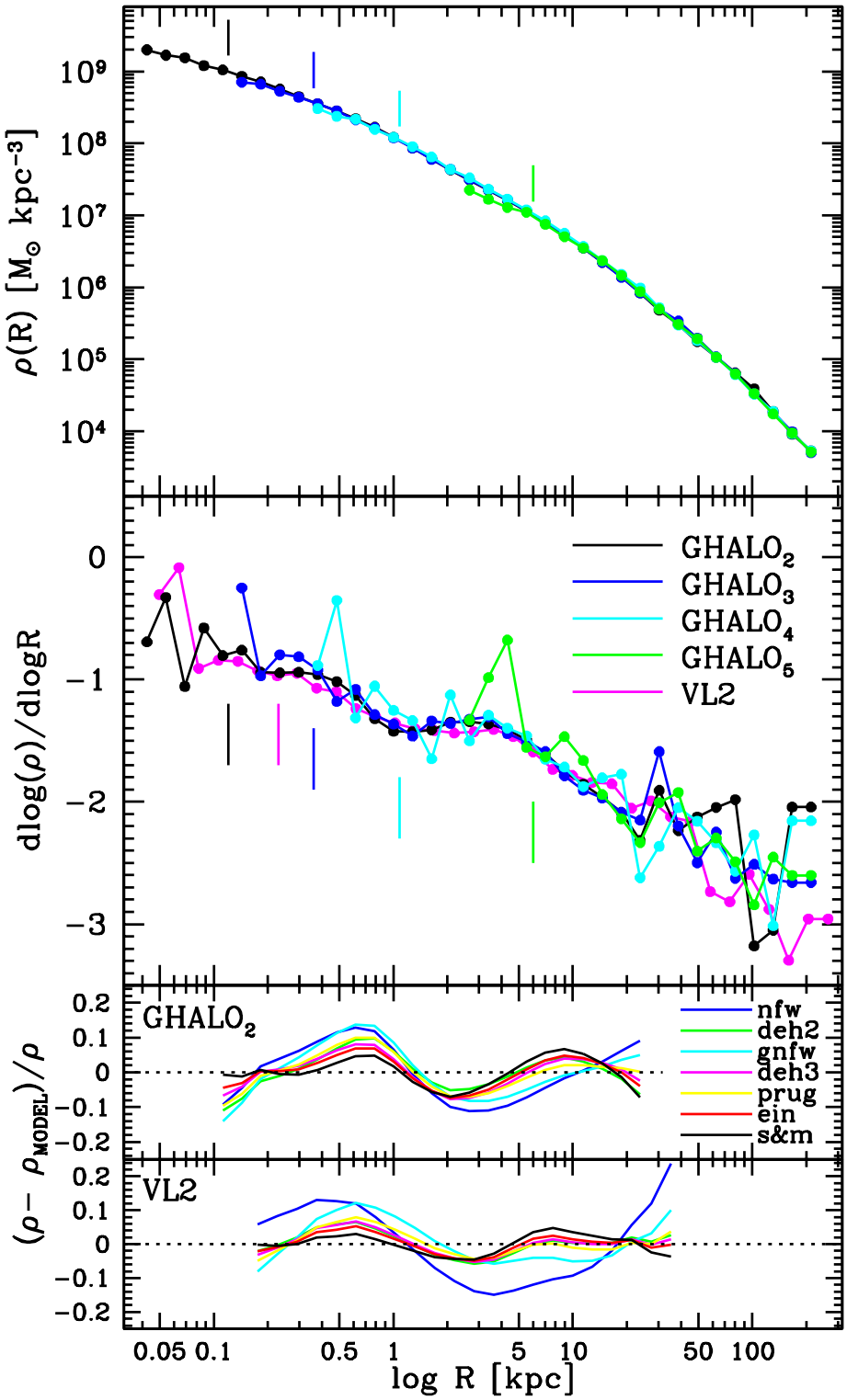


Figure 7.2: The upper panel shows the density profile of $GHALO_2$ and its lower resolution realizations as well as the density profile of the VL2 simulation in magenta. The convergence radius at each step in resolution is easily seen (indicated by the tick marks). The lower panel shows the residuals of the $GHALO_2$ simulation with respect to 2-parameter fitting functions: NFW (blue) and Dehnen-McLaughlin (green); as well as 3-parameter fitting functions: S&M-profile (black), Einasto (red), Generalized NFW (cyan), Dehnen-McLaughlin (magenta), Prugniel-Simien (yellow).

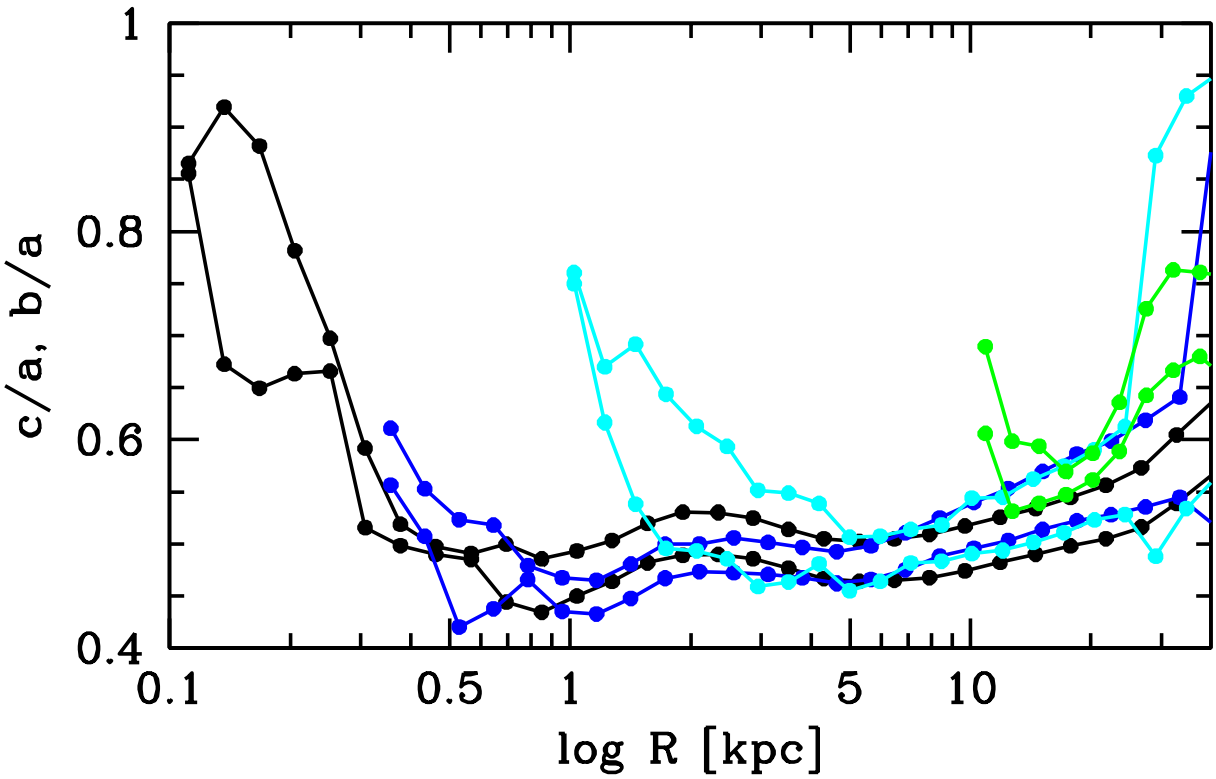


Figure 7.3: Shape parameters for $\mathrm{GHALO}_{2,3,4,5}$.

profiles, although the best fit parameters vary. Due to the prolate shape the density profile parameters for the short axis are similar to the ones presented in Table 1.

7.3.3 Phase-Space Density Profile

It has been pointed out [199, 50, 52] that the phase-space density (PSD) proxy, $\rho\sigma^{-3}$ vs R is a power-law for CDM haloes, and several new fitting functions for the density profile have been proposed using this fact as a starting point such as the Dehnen-McLaughlin models. When averaged in shells, $\rho(2\pi\sigma^2)^{-3/2}$ is remarkably well fit by a power-law with slope of -1.84 as shown in Figure 4. However, it is interesting to compare this spherically averaged estimate with the true 6-dimensional PSD.

The code EnBiD [178] has improved on earlier work by Ascasibar and Binney [10] in calculating better estimates of the 6-dimensional phase-space volume occupied by each particle and hence the PSD. Taking the mean EnBiD PSD in logarithmic shells we see that the closest subhalo at 1.8 kpc stands out prominently and subhalos at larger radii begin to dominate the mean. Using a method based on a 6-dimensional Voronoi tessellation Arad, Dekel and Klypin [9] also showed that the subhalos form a dominant contribution to the phase-space density. This feature of using the EnBiD PSD can be turned to great advantage in identifying subhalos and other substructures such as phase-space streams. However, removing the effect of subhalos with $f_{\mathrm{EnBiD}} > 100\ M_\odot \mathrm{kpc}^{-3}(\mathrm{km/s})^{-3}$ from the mean, we extend the mean background PSD out to much larger radii as shown in Figure 4. By removing streams, with $f_{\mathrm{EnBiD}} > 0.4 M_\odot \mathrm{kpc}^{-3}(\mathrm{km/s})^{-3}$, we can extend this to at least 40 kpc.

We find that the true radial PSD profile estimated with EnBiD does not follow such a perfect power law and shows a steeper slope (roughly -2) than the $\rho\sigma^{-3}$ estimator. The EnBiD mean estimate and $\rho(2\pi\sigma^2)^{-3/2}$ are in agreement from about 0.2 to 2 kpc but the meaning of the power-law behaviour of $\rho(2\pi\sigma^2)^{-3/2}$ is unclear given that inside of 0.2 kpc it is under-resolved and outside of 2 kpc a large contribution comes from the substructure. A further concern is the considerable variation of $\rho\sigma^{-3}$ about a spherical shell of the prolate inner halo, which makes it remarkable that we obtain the same power-law slope as originally found by [199] despite the averaging that is taking place. This also explains the good performance of the Dehnen-McLaughlin 2 and 3-parameter models at fitting the density profile.

From about 2 to 40 kpc the $\rho\sigma^{-3}$ estimator is somewhat enhanced due to the presence of substructure, while inside of 0.1 kpc the EnBiD–mean continues to resolve the power-law behaviour of the profile.

7.4 Conclusions

The GHALO_2 simulation has achieved an unprecedented spatial and mass resolution within a CDM halo, resolving thousands of subhalos within a radius corresponding to the galactic disk and a rich phase-space structure of streams beyond a radius of ~ 8 kpc. Whilst there are more detailed analyses of this simulation in progress, we have reported here on the global inner properties of density and phase-space density profiles and halo shape. Using a sequence of simulations of the same halo at difference resolutions, from 10^5 – 10^9 particles, we confirm that the convergence radii for the density profile and shape scales as $N_{\rm vir}^{-1/3}$. The logarithmic slope of the radial density profile is close to a power law, gradually turning over to a slope of -0.8 at our innermost resolved region (0.05% of $R_{\rm vir}$). We have proposed a new two parameter fitting function that has a linearly varying logarithmic gradient which provides the best fit to the inner part of the GHALO and VL2 haloes. A larger sample of haloes, such as Bullock et al. [32] and Maccio' et al. [122], would be required to determine if this functional form provides a universal fit. We find that the convergence radius of the density is a factor of three smaller than the convergence of halo shape. GHALO is prolate, yet becomes spherical within a region where orbits are most likely innacurately followed due to the effects of finite particle number, relaxation and softening.

All functional forms fit to density profiles, whether 2 or 3 parameters are empirical fits, even those based on properties (the Dehnen-McLaughlin) of the phase-space density profile whose origin is still poorly understood. Therefore the only current confidence can be given to those profiles which have been fit to the highest resolution simulations and over the widest range of halos encountered in N-body simulations. Clearly these two criteria are in conflict since simulations at the resolution of GHALO are too expensive to allow a broader study. Therefore, the results presented here should be considered as guides only, whose generality remains to be tested. Never the less we can consider economy of parameters and simplicity of functional form as guiding principles in the search for suitable profile functions to describe the end state of gravitational collapse. All the profiles we fit here (Table 1 and residuals in Figure 2) meet these subjective criteria, having at most 3 free parameters and simple functional forms.

While the phase-space density estimated by $\rho\sigma^{-3}$ is observed to follow a power law in

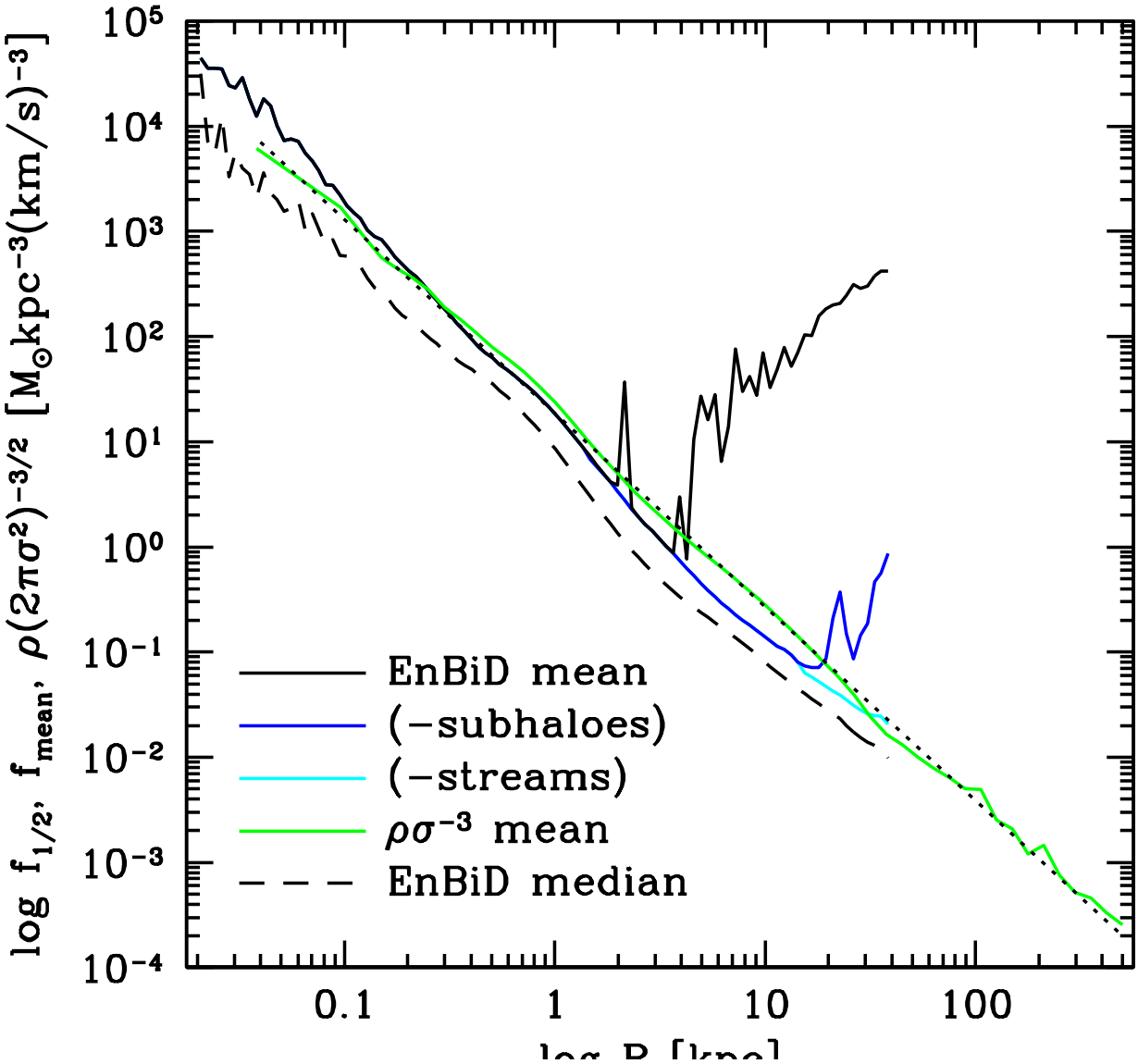


Figure 7.4: The phase-space density profile of the main halo, measured in several different ways is shown. The solid green line shows the traditional $\rho(2\pi\sigma^2)^{-3/2}$ averaged in shells. The solid black and dashed black curves shows the mean and median EnBiD phase-space density estimator [178] for the particles in logarithmic shells extending out to 40 kpc. The blue and cyan curves show the mean EnBiD phase-space density profile, but where the contribution from subhalos (blue) and subhalos+streams (cyan) has been excluded. Despite the effects of substructure the $\rho\sigma^{-3}$ profile is remarkably well fit by a power-law with slope -1.84 shown as the dotted black line.

radius of slope -1.84, its meaning is less clear since at small radii it is limited by resolution of the estimator and at larger radii it becomes dominated by subhalos. Using the more sophisticated EnBiD PSD estimator we find that the radial profile is steeper with an index of about -2, but that it is not as perfect a power-law as seen in $\rho\sigma^{-3}(r)$.

As a final comment, we note that in large galaxies, the inner structure and shape of the dark matter halo has likely been altered over time by the baryons via a range of physical effects, including dissipation, energy transfer from sinking massive objects, binary black holes, bar-halo interactions, turbulent gas motions and more. Simulations that follow the baryonic components together with the dark matter will resolve these additional questions in the coming years.

Acknowledgments

We are grateful to the Barcelona Supercomputing Center for their generous allocation of resources on the Marenostrum supercomputer and to Jose Maria Ibañez for his enthusiastic support of this project. Support for this work was also provided by the Spanish Ministry grant MEC AYA2007-67752-C03-02 (V.Q.) and by NASA through grants HST-AR-11268.01-A1 and NNX08AV68G (P.M.) and Hubble Fellowship grant HST-HF-01194.01 (J.D.).

Chapter 8

Matter power spectrum and the challenge of percent accuracy

Journal of Cosmology and Astroparticle Physics, Issue 04, article id. 047 (2016)

Aurel Schneider[1], Romain Teyssier[1], Doug Potter[1], Joachim Stadel[1], Julian Onions[2], Darren S. Reed[1], Robert E. Smith[3], Volker Springel[4,5], Frazer R. Pearce[2] & Roman Scoccimarro[6]

[1]Institute for Computational Science, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland

[2]School of Physics and Astronomy, University of Nottingham, Nottingham NG7 2RD, United Kingdom

[3]Department of Physics and Astronomy, University of Sussex, Brighton, BN1 9QH, United Kingdom

[4]Heidelberger Institut für Theoretische Studien, 69118 Heidelberg, Germany

[5]Zentrum für Astronomie der Universität Heidelberg, ARI, 69120 Heidelberg, Germany

[6]Center for Cosmology and Particle Physics, Department of Physics, New York University, NY 10003, New York, USA

Future galaxy surveys require one percent precision in the theoretical knowledge of the power spectrum over a large range including very nonlinear scales. While this level of accuracy is easily obtained in the linear regime with perturbation theory, it represents a serious challenge for small scales where numerical simulations are required. In this paper we quantify the precision of present-day N-body methods, identifying main potential error sources from the set-up of initial conditions to the measurement of the final power spectrum. We directly compare three widely used N-body codes, `Ramses`, `Pkdgrav3`, and `Gadget3` which represent three main discretisation techniques: the particle-mesh method, the tree method, and a hybrid combination of the two. For standard run parameters, the codes agree to within one percent at $k \leq 1\ h\,\mathrm{Mpc}^{-1}$ and to within three percent at $k \leq 10\ h\,\mathrm{Mpc}^{-1}$. We also consider the bispectrum and show that the reduced bispectra agree at the sub-percent level for $k \leq 2\ h\,\mathrm{Mpc}^{-1}$. In a second step, we quantify potential errors due to initial conditions, box size, and resolution using an extended suite of simulations performed with our fastest code `Pkdgrav3`. We demonstrate that the simulation box size should not be smaller than $L = 0.5\ h^{-1}\mathrm{Gpc}$ to avoid systematic finite-volume effects (while

much larger boxes are required to beat down the statistical sample variance). Furthermore, a maximum particle mass of $M_{\rm p} = 10^9\ h^{-1}{\rm M}_\odot$ is required to conservatively obtain one percent precision of the matter power spectrum. As a consequence, numerical simulations covering large survey volumes of upcoming missions such as `DES`, `LSST`, and `Euclid` will need more than a trillion particles to reproduce clustering properties at the targeted accuracy.

8.1 Introduction

In the last decades, cosmology has entered the high precision regime owing to ever more accurate measurements of the Cosmic Microwave Background (CMB). The statistical information of the CMB sky is, however, intrinsically limited, while large scale structures contain a great wealth of modes which can be exploited, provided nonlinear structure formation is well understood. Next-generation galaxy and weak lensing surveys such as `DES`[1], `LSST`[2], and `Euclid`[3] require percent accurate modelling of the matter power spectrum up to wave numbers of $k \sim 10\ h\,{\rm Mpc}^{-1}$ in order to fully exploit their constraining power for cosmology [98, 115].

Standard perturbation theory gives accurate results up to $k \sim 0.1\ h\,{\rm Mpc}^{-1}$, while numerical simulations are indispensable at higher wave numbers [34, 65]. Pure dark matter simulations based on N-body techniques are believed to be accurate up to about $k \sim 0.5\ h\,{\rm Mpc}^{-1}$, beyond which baryonic feedback from active galactic nuclei (AGN) must be included [46, 168].

In this paper we focus on the matter power spectrum from collisionless N-body simulations, ignoring all hydrodynamical effects. Although strictly not valid at small scales, this approach is currently the only option for precision cosmology as the relevant AGN feedback mechanism is not well understood, and is poorly constrained by observations. A potential way forward is to study AGN feedback with high resolution simulations of small cosmological volumes and to parametrise the effects on the matter power spectrum. Cosmological parameter estimation can then be carried out on the basis of N-body simulations plus additional free model parameters accounting for the AGN contribution [135, 168].

Comparison studies of N-body codes and subsequent analysis tools have been performed in the past. The first investigations of different N-body techniques was carried out in Ref. [61] more than thirty years ago. More recently, the authors of Ref. [145] compared high-redshift power spectra and halo abundances from mesh- and particle-based techniques, reporting significative differences at small scales. The first detailed code comparison study including six gravity codes was carried out by Ref. [84]. In terms of the power spectrum, the authors reported agreement of roughly ten percent between particle codes up to $k \sim 10$, while mesh codes deviated already at smaller wave-numbers due to incomplete resolution. Three years later, another comparison project with 10 gravity codes was carried out by Ref. [85], showing further improvement in code agreement and stating roughly one percent differences for the power spectrum below $k \sim 1$ h/Mpc. At larger wave-numbers they observed growing discrepancy between mesh and particle codes which exceeded ten percent at $k \sim 10$ h/Mpc, while both methods were shown to agree to about five percent among each

[1] `www.darkenergysurvey.org`
[2] `www.lsst.org/lsst`
[3] `sci.esa.int/euclid`

other. Finally, Ref. [87] reinvestigated the difference between mesh and particle methods using simulations with larger, cosmological viable box sizes and particle numbers. They confirmed the reported percent agreement up to $k = 2$ h/Mpc.

In the present paper we build upon these past efforts and compare three gravity codes representing the most widely used discretisation techniques. We thereby use an unprecedented setup in terms of box size and particle resolution to allow for a code comparison free of systematic effects. This is confirmed in the second part of the paper were we show that potential systematics from initial conditions, box size, and particle numbers are below the percent error condition.

Sec. 8.2 is devoted to the code comparison, focusing on the auto and cross power spectra. In Sec. 8.3 we take a critical look at the simulation pipeline and investigate the accuracy of the initial conditions as well as potential finite volume and resolution effects. A summary of the results including a list of requirements to obtain percent accuracy of the matter power spectrum is presented in Sec. 8.4. In the Appendices, we investigate modifications of the code parameters (Appendix 8.5) and we present a code comparison beyond the power spectrum (Appendix 8.6).

8.2 Code comparison

The first part of this paper is about comparing N-body codes with respect to the precision requirements of upcoming galaxy and weak lensing surveys. Our study is mainly focused on the auto power spectrum which is the prime statistical measure in cosmology. Additionally, we investigate phase-shifts in Fourier space by cross-correlating the results of different codes in order to further quantify the spatial disagreement between density fields from different N-body techniques.

8.2.1 N-body codes

The gravitational N-body technique is the standard tool to simulate the nonlinear Universe, yielding accurate results at scales where hydrodynamical effects are subdominant. Most N-body codes are either based on a particle-mesh method, a tree algorithm, or a hybrid combination of the two. In this paper, we compare the codes `Ramses`, `Pkdgrav3`, and `Gadget3`, which represent each of these three approaches and are widely used in the astrophysics and cosmology community.

The comparison is performed by running a simulation of box size $L = 500\ h^{-1}$Mpc and resolution of $N = 2048$ particles per dimension with each of the three codes, starting from the exact same initial conditions and using the standard run parameters described below. The initial conditions are based on second order Lagrangian perturbation theory (2LPT), generated at redshift 49 with a modified version of the IC code from [39, 172]. For the cosmological parameters, we use Planck values, i.e., $\Omega_m = 0.3071$, $\Omega_\Lambda = 0.6929$, $\Omega_b = 0.0483$, $h = 0.6777$, $n_s = 0.9611$, and $\sigma_8 = 0.8288$ [1]. The measurement of the power spectra is performed at exactly the same redshifts and with the same analysis tool (using the triangular shaped cloud scheme for the mass assignment). In this way, we carefully

avoid all other potential sources of error and directly compare effects due to the gravity calculations of the codes.

We now briefly present the three codes and give details about the run parameters for the comparison:

- The N-body and hydrodynamical code `Ramses` [201] is based on a particle-mesh technique and uses adaptive mesh refinement for high density regions. The code is mainly used for hydrodynamical simulations in a cosmological context [142, 3] but it has also been employed for pure dark matter N-body runs in the past [202, 158]. For the comparison, we apply a coarse-level grid with refinement level $\ell_{\rm min} = 12$, corresponding to 2048^3 coarse cells. New refinements are triggered on a cell-by-cell, recursive basis when a cell collects more than 8 particles. Using this strategy we reach a maximum level of refinement $\ell_{\rm max} = 18$, corresponding to a spatial resolution of 2 h^{-1}kpc. We employ adaptive, level-by-level time-stepping, with a time step size set smaller than the local free fall time, and by the requirement that a particle cannot move more than half a cell within one time step. The convergence criterion for the Poisson solver, defined as the ratio of the residual L^2-norm to the right-hand side L^2-norm, is set to $\epsilon = 10^{-4}$.

- The gravity code `Pkdgrav3` [an earlier version of which is described in 195] is based on a binary tree algorithm using fifth order fast multipole expansion of the gravitational potential (using cell-cell interactions making it an $\mathcal{O}(N)$ gravity calculation method). Periodic boundaries conditions are calculated with the Ewald summation technique, requiring very little data movement while exposing a high degree of parallelism. `Pkdgrav` has been extensively used for N-body simulations in the past, mainly in the context of cosmological zoom simulations such as *Via Lactea* [58] and *Ghalo* [196]. The current version of `Pkdgrav` includes GPU acceleration for all force calculations, leading to a significant speed-up with respect to previous versions. In this paper, we use the run parameters $\varepsilon = 0.02\, l_{\rm mean}$ (where $l_{\rm mean}$ is the mean particle separation) and $\theta = 0.7$ ($\theta = 0.55$ above redshift two) for softening and tree opening criteria. The adaptive time-stepping is parametrised in the standard way, i.e $dt_i = \eta\sqrt{\varepsilon/|a_i|}$ with $\eta = 0.15$ (where a_i is the acceleration of particle i). `Pkdgrav3` also has a more sophisticated time-stepping criterion based on an estimation of the local dynamical time.

- The tree-particle-mesh code `Gadget3` applies a uniform particle-mesh method at large scales plus a first order oct-tree technique at small scales [see 192, for a description of an earlier version of the code]. `Gadget` is extensively used in many contexts and is most known for the *Millennium* suite of cosmological simulations [193], as well as the *Aquarius* zoom simulations [194]. For the comparison, we use a comoving Plummer-equivalent softening length of $\varepsilon = 10\ h^{-1}$kpc and the code's relative tree opening criterion with a tolerance value of $\alpha = 0.005$ for the gravitational force accuracy [see 192, for more information]. Furthermore, we adopt a time integration parameter corresponding to $\eta = 0.22$ for the time-stepping criterion used above in `Pkdgrav3`. The long-range particle-mesh forces are calculated with a 2048^3 Fourier grid.

All simulations used of this comparison were performed on the hybrid CPU/GPU cluster *Piz Daint* at the Swiss National Supercomputing Centre (CSCS). The total run-time for the

three codes is 94 352 node-hours for `Ramses`, 34 524 node-hours for `Gadget3`, and 1632 node-hours for `Pkdgrav3` (the former two codes were run on 512 nodes using CPU only, while the latter was run on 128 nodes using full GPU acceleration for the force calculations). On each node of *Piz Daint* there are 8 CPU-cores and one Nvidia Tesla K20X GPU accelerator.

8.2.2 Definitions

Before discussing differences between the gravity codes, we give a brief definition of the power spectrum and cross power coefficient (definitions of the propagator and the bispectrum can be found in Appendix 8.6). Let us first assume we have two density fields $\delta_X(\mathbf{k})$ and $\delta_Y(\mathbf{k})$ in Fourier space. The power spectrum $P_{XY}(k)$ is then defined as (see e.g. Ref. [21])

$$\langle \delta_X(\mathbf{k}) \delta_Y(\mathbf{k}') \rangle \equiv \delta_D(\mathbf{k} + \mathbf{k}') P_{XY}(k), \tag{8.1}$$

where $\delta_D(\mathbf{x})$ is the three dimensional Dirac delta function. Eq. (8.1) defines both the auto and the cross power spectrum, which we now briefly discuss. The auto power spectrum is given by

$$P(k) \equiv P_{XX}(k) \tag{8.2}$$

and provides a measure of the density *amplitudes* at different k-modes. The cross power spectrum P_{XY} (with $X \neq Y$), on the other hand, also measures the *phase* differences between density fields. It is convenient to the define the cross power coefficient (e.g. [149, 200, 175])

$$r_{XY}(k) \equiv \frac{P_{XY}(k)}{\sqrt{P_{XX}(k) P_{YY}(k)}}, \tag{8.3}$$

which only contains information about phase shifts while all amplitudes are factored out. This becomes evident if we split the perturbation field into an amplitude and a phase component, i.e. $\delta_X(\mathbf{k}) = \Delta_X(\mathbf{k}) \exp[i\phi_X(\mathbf{k})]$ (see e.g. Ref. [35]). The fact that the cross power coefficient measures the spatial shifts between density fields makes it an interesting alternative indicator for a code comparison.

8.2.3 Auto power spectrum

Analysing the accuracy of the auto power spectrum $P(k)$ from numerical simulations is the main goal of this paper. The measurement of $P(k)$ is performed with fully mpi-parallel Fast Fourier Transform (FFT) using the triangular shaped cloud (TSC) method to assign particles on the grid[4].

The resulting power spectra are shown in Fig. 8.1, where different panels correspond to different redshifts $z = 3.8$, 2, 1.0, and 0.0. The green lines refer to `Pkdgrav3`, the red lines to `Gadget3`, and the blue lines to `Ramses`[5]. One percent agreement (grey shaded area) between the different codes is obtained up to $k \sim 1\, h\,\mathrm{Mpc}^{-1}$ over all redshifts, as illustrated by the vertical dashed line. In the highly nonlinear regime from $k = 1$ to $10\, h\,\mathrm{Mpc}^{-1}$, the

[4]In order to avoid smearing effects, we normalise the density contrast in k-space with the Fourier transform of the assignment window (see e.g. [102, 42, 36] for more information).

[5]We have chosen the blue lines to act as reference, solely because it lies between the green and red lines at $z = 0$, therefore improving the readability of the plots.

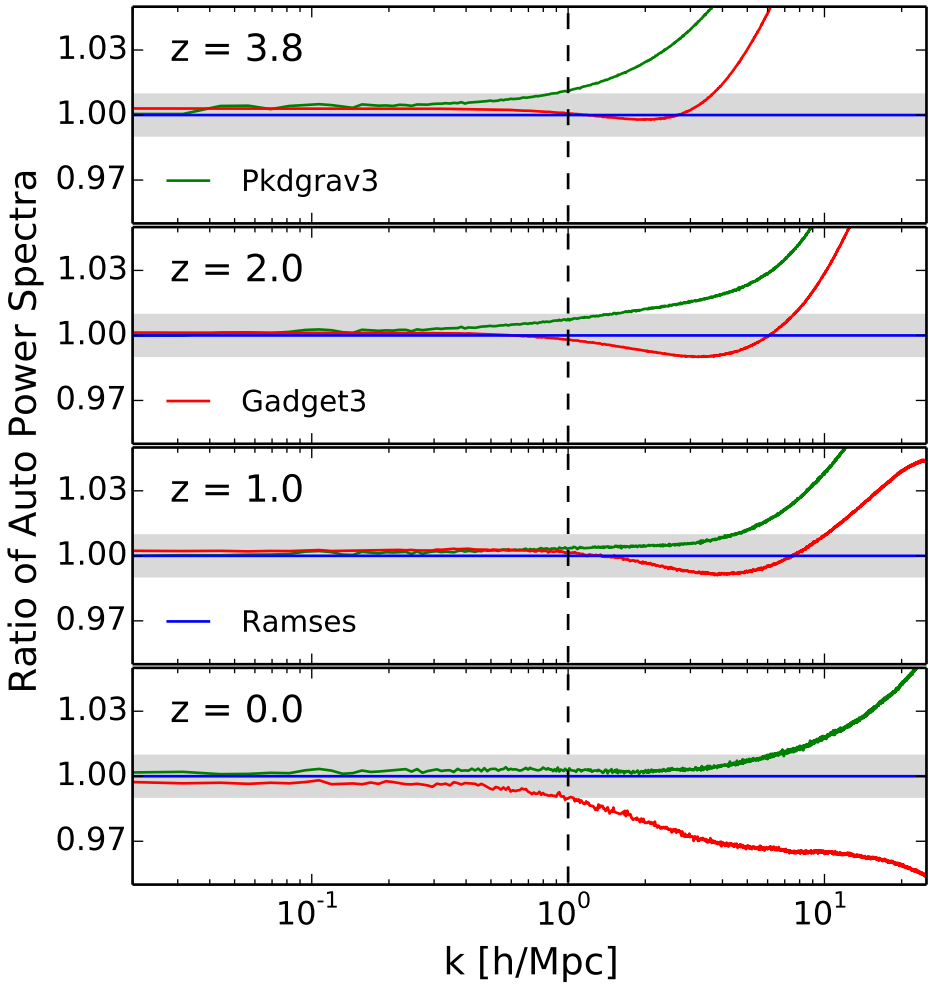


Figure 8.1: Comparison of auto power spectra from the three different N-body codes at different redshifts. Green lines correspond to `Pkdgrav3`, red lines to `Gadget3`, and blue lines to `Ramses` (reference lines). One percent agreement (indicated by the grey band) is obtained for $k \leq 1\ h\,\mathrm{Mpc}^{-1}$ (dashed vertical line).

agreement between codes is at the three percent level for $z = 1$ and below[6]. At higher redshifts, the discrepancy grows, reaching about five percent at $z = 2$ and ten percent at $z = 3.8$.

The agreement between the codes is significantly better than in previous code comparison projects by Heitmann et al. [85, hereafter H08] and Heitmann et al. [87, H10] illustrating the progress in code development over the last five years. At very large scales we obtain maximal differences of ~ 0.4 percent between codes, with respect to ~ 3 percent in H08 and ~ 2 percent in H10[7]. At small scales beyond $k \sim 1$ h/Mpc, the prominent systematic offset between PM-codes and tree-codes visible in H08 (with more than 10 percent difference at $k \sim 10\ h\,\mathrm{Mpc}^{-1}$) has now entirely disappeared.

[6]While `Ramses` and `Pkdgrav3` show percent agreement until $k \sim 7\ h\,\mathrm{Mpc}^{-1}$, `Gadget` slightly deviates at $k \sim 1\ h\,\mathrm{Mpc}^{-1}$.

[7]The better agreement between `Pkdgrav3` and `Gadget3` is (at least partially) because of a new implementation of the periodic boundary conditions in `Pkdgrav`.

The relatively large difference between `Pkdgrav3`/`Gadget3` and `Ramses` at z=3.8 can be explained by the fact that AMR codes require many more particles to resolve haloes. As we will see in the following sections, the power spectrum is dominated by group-sized haloes at redshift zero, which are well resolved in our simulations. At high redshift, however, the signal stems from considerably smaller structures which are better resolved with the tree-codes than with an AMR technique. Higher resolution of the AMR grid is required to remedy this.

The results of the code comparison are based on the standard run parameters described above. These parameters have never been systematically tested in the context of large-scale cosmological simulations, but they emerged via many different code applications in the past. However, finding more optimal code parameters is non-trivial because the true power spectrum is not known *a priori* . Parameters cannot simply be tuned to achieve maximal agreement between codes since this could lead to convergence towards the wrong answer.

It is nevertheless important to quantify the dependency of the run parameters on the resulting power spectrum. Only results which are insensitive to the choice of code parameters can be trusted. In Appendix 8.5 we study the effects of the most sensitive code parameters, which are the size of the PM grid for `Gadget3` as well as the time-stepping criterion for all three codes. We conclude that reasonable variations of these parameters lead to sub-percent effects on the power spectrum below $k \sim 10\ h\,\mathrm{Mpc}^{-1}$, smaller than the observed differences between codes visible in Fig. 8.1.

Summing up, the code comparison suggests that the consensus between different N-body techniques is good, however not quite good enough for the targeted percent accuracy up to $k \sim 10\ h\,\mathrm{Mpc}^{-1}$. Further improvements to the codes will not be easily achievable as the correct solution for the matter power spectrum is not known. A common effort of the community is required to converge towards a generally accepted solution. In order to encourage further comparison of N-body codes, we release the IC file used here plus all power spectrum measurements on `www.ics.uzh.ch/~aurel/`.

8.2.4 Cross power spectrum

The cross power coefficient r_{XY} (defined in Eq. 8.3) quantifies the spatial shifts between two density fields and is therefore a sensitive statistical measure to compare N-body codes. While the auto power spectrum only gives information about the amplitude of perturbations, the cross power coefficient measures the relative phase-shifts for any given k-mode. The cross power spectrum is obtained by separately Fourier-transforming the two density fields from different N-body codes.

In Fig. 8.2 we plot the cross power coefficients based on density fields from `Gadget3-Pkdgrav3` (brown), `Gadget3-Ramses` (magenta), and `Pkdgrav3-Ramses` (cyan) at redshift 2 (top) and redshift 0 (bottom). At the largest scales ($k \lesssim 0.5\ h\,\mathrm{Mpc}^{-1}$), the results are in perfect agreement, which can be explained by the fact that the cross power coefficient is independent of the growth factor and therefore insensitive to errors related to global time-integration. At smaller scales and especially at low redshift, the deviations are considerably larger than the differences observed in the power spectrum. This is due to the effect of

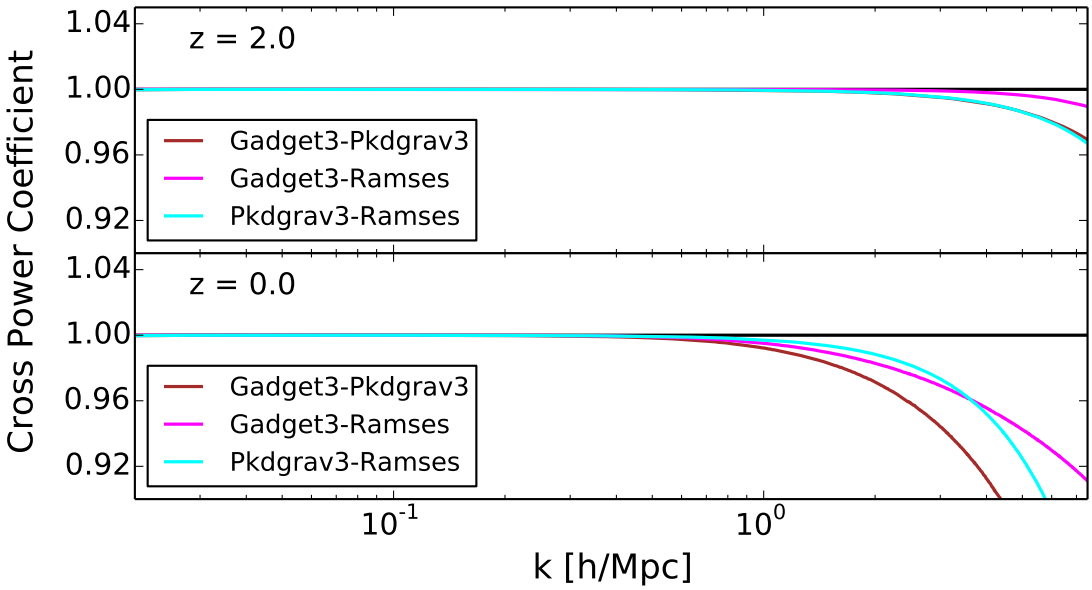


Figure 8.2: Cross power coefficient (as defined in Eq. 8.3) at redshift two (top) and redshift 0 (bottom). Brown, magenta, and cyan lines correspond to the combinations of density fields from `Gadget3-Pkdgrav3`, `Gadget3-Ramses`, and `Pkdgrav3-Ramses`.

gravity which magnifies deviations of phases over time, something that is clearly visible in Fig. 8.2.

In general, the cross power coefficients from different code combinations are in good agreement with each other. At redshift 2, there are no visible phase-shifts up to $k \sim 2$ h/Mpc. At smaller scales some small differences start to appear, while the density fields from `Gadget3` and `Ramses` seem to agree somewhat better with each other than with the density field from `Pkdgrav3`. At redshift zero, phases-shifts start to be visible above $k \sim 0.5$ $h\,\mathrm{Mpc}^{-1}$. The largest differences are observed between the density fields of `Gadget3` and `Pkdgrav3`, which is surprising given the fact that they use similar numerical techniques at small scales.

In summary, we want to highlight the extremely good agreement of the cross power coefficients below $k \sim 0.5$ h/Mpc suggesting vanishing force errors at large scales. As a consequence, the sub-percent differences visible in power spectrum at large scales (see Fig. 8.1) have to come from slightly different growth factors and are therefore potentially stemming from small inaccuracies in the global time-integration schemes. Finally, we want to stress that both the auto and cross power spectra shown in this paper only provide measures for the relative differences between codes but do not indicate which one of the three codes is most accurate.

8.3 Testing the N-body pipeline

Potential inaccuracies of numerical simulations are not restricted to the N-body code but can stem from the initial conditions, limited box-size or physical resolution. Each of these

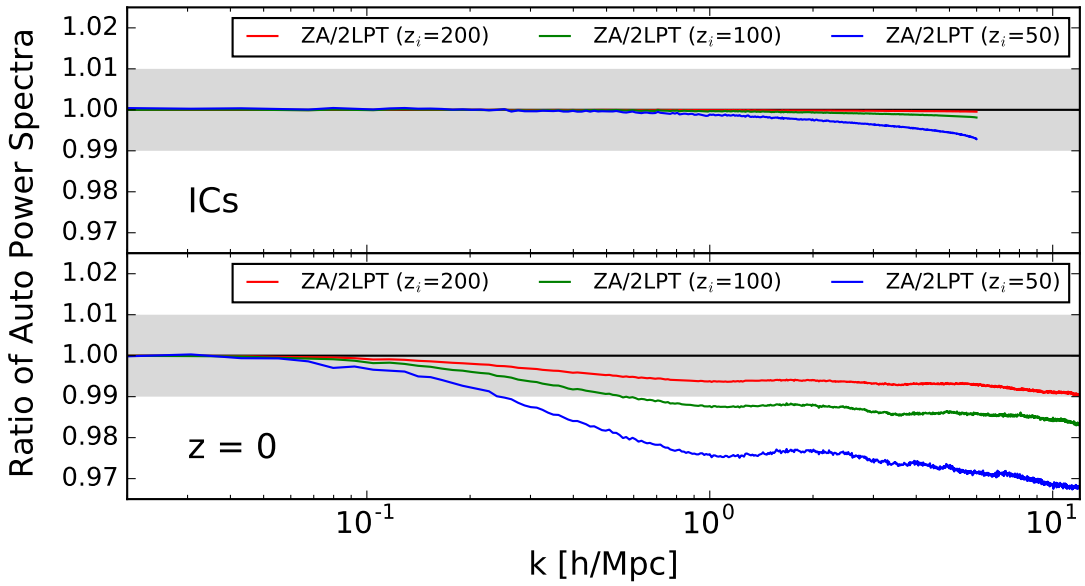


Figure 8.3: Ratios of power spectra from simulations with ZA and 2LPT initial conditions and different starting redshifts (z_i). The top panel shows measurements at redshift z_i, the bottom panel at redshift zero. One percent agreement is illustrated by the grey band.

sources of error has been extensively studied in the past (see e.g. [87, 158]). Here, we reanalyse potential effects from initial conditions, box-size and resolution with the focus on the requirement of sub-percent errors.

8.3.1 Initial Conditions

Initial conditions of cosmological simulations are generated as a random realisation of a (Gaussian) density field, based on either first or second order Lagrangian perturbation theory. The density field is usually discretised in form of aligned particles on an initial grid, where small displacements account for the initial perturbations.

The redshift of the initial conditions has to be chosen with care. It should lie in a range where all resolved perturbations are large enough to dominate numerical noise, but still small enough to be accurately described by perturbation theory. It has been shown in the past that it is advantageous to use second order Lagrangian perturbation theory (2LPT) with respect to the simpler first order or Zel'dovich approximation (ZA), as it allows for smaller starting redshifts, further away form the noise dominated high-redshift regime [171, 39, 99, 159].

We study the effects of the initial conditions on the power spectrum at redshift zero by running simulations with $L = 512\ h^{-1}$Mpc and $N = 1024$ particles per dimension with the N-body code `Pkdgrav3`. The initial conditions are generated with `MUSIC` [80], using both the ZA and 2LPT approach at different starting redshifts (z_i).

The resulting effects on the power spectrum are illustrated in Fig. 8.3. In the top panel, we show the ratios between ZA and 2LPT directly measured in the ICs at the corresponding

starting redshifts of $z_i = 200$ (red line), $z_i = 100$ (green line), and $z_i = 49$ (blue line). The differences between ZA and 2LPT are at sub-percent level (converging towards large z_i) and limited to high wave numbers above $k \sim 1\ h\,\mathrm{Mpc}^{-1}$. In the bottom panel, we show the same ratios now measured at redshift zero. The differences between ZA and 2LPT have grown substantially affecting wave number beyond $k \sim 0.1\ h\,\mathrm{Mpc}^{-1}$. This behaviour is in agreement with previous findings [39, 108, 117].

Fig. 8.3 suggests that a starting redshift $z_i \gtrsim 200$ is required to obtain percent accuracy with ZA initial conditions. Such high starting redshifts are prone to numerical problems, since N-body codes do not deal well with extremely small initial density perturbations. At what redshift numerical effects become a problem depends on the code and the run parameters. Based on a study involving `Pkdgrav2` and `Gadget2`, Reed et al. [159] concluded that the initial redshift and the redshift of typical halo formation should not differ by more than a factor of fifty. For the cosmological boxes investigated here, most haloes form around redshift two [see for example 131] which results in the requirement $z_i \lesssim 100$[8].

In agreement with previous results, we conclude that initial conditions with 2LPT should be used consistently for cosmological simulations. They are significantly more accurate than ZA initial conditions and they allow lower starting redshifts, thus decreasing the run-time of simulations.

8.3.2 Box size and resolution

A careful setup of simulations in terms of box size and particle numbers is crucial in order to obtain one percent agreement in the power spectrum. Small boxes tend to suffer from sample variance and missing large-scale modes, while large boxes might not have enough resolution to capture the very nonlinear scales.

It is straight-forward to determine the expected statistical (Gaussian) error which consists of a sample variance and a shot-noise contribution and is given by

$$\Delta P(k) = \left(\frac{2}{\Delta N_\mathrm{m}}\right)^{1/2} \left[P(k) + P_\mathrm{sn}\right], \tag{8.4}$$

where $\Delta N_\mathrm{m} = L^3 k^2 \Delta k/(2\pi^2)$ is the number of modes per k-bin and $P_\mathrm{sn} \equiv (L/N)^3$ is the Poisson shot-noise. From Eq. (8.4) it becomes obvious that the sample variance increases towards larger scales. Enforcing sub-percent statistical errors and assuming $\Delta k = 2\pi/L$ results in the condition

$$L \gtrsim \frac{250}{k} \tag{8.5}$$

for the box length L. This means that a minimal box length of $L = 2.5\ h^{-1}\mathrm{Gpc}$ is required to beat down the sample variance below the percent level for k-modes above $k \sim 0.1\ h\,\mathrm{Mpc}^{-1}$. The shot noise contribution P_sn, on the other hand, becomes important at the smallest scales. In this paper we subtract Poisson shot-noise from all measured power spectra and we indicate the scale above which shot-noise contributes at more than one percent to the total power spectrum.

[8] Recent tests with `Pkdgrav3` show that high-redshift errors can be reduced by choosing a smaller tree-opening parameter during the first gravity steps. This could potentially allow to shift the starting redshift to higher values for the same precision requirements.

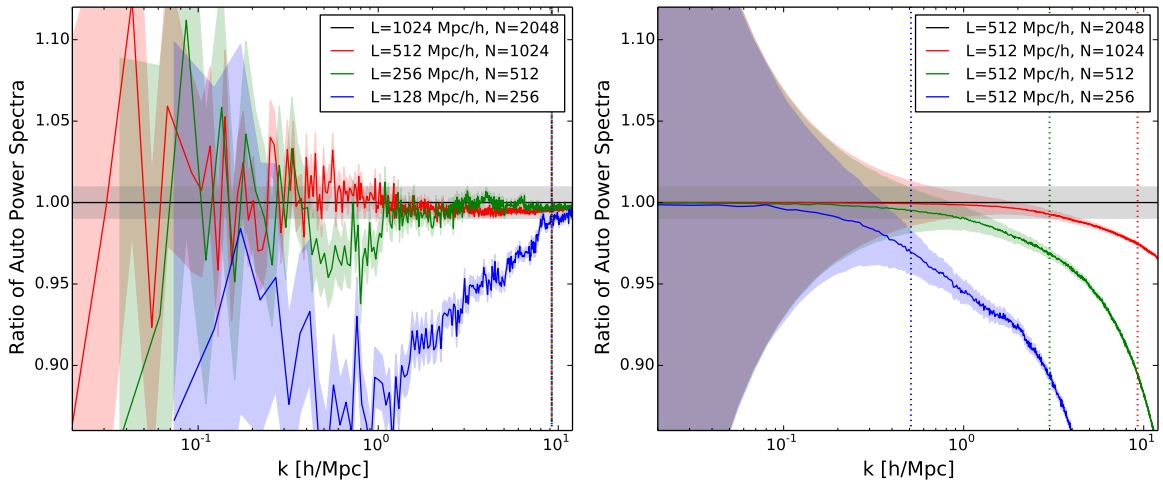


Figure 8.4: Investigating box-size and resolution effects on the power spectrum. *Left panel:* Same physical resolution and increasing box size: L=128 h^{-1}Mpc (blue), L=256 h^{-1}Mpc (green), L=512 h^{-1}Mpc (red), L=1024 h^{-1}Mpc (black, reference line). *Right panel:* Same box size (L=512 h^{-1}Mpc) and increasing number of particles (per dimension): N=256 (blue), N=512 (green), N=1024 (red), and N=2048 (black, reference line). The coloured areas quantify the statistical errors from sample variance, while the grey shaded band highlights the range of percent accuracy. Dotted vertical lines indicate where the shot-noise contribution exceeds one percent.

Next to statistical errors there are systematical effects due to finite volume and resolution of the simulation setup as well as the nonlinear nature of gravity. These errors are more difficult to quantify and we will focus on providing estimates of how they can be minimised to sub-percent level.

We use `Pkdgrav3` to run a suite of numerical simulations with varying box size and particle number. Volume effects on the power spectrum are investigated by comparing simulations with the same mass resolution and different box sizes. Effects due to particle numbers are studied with runs of constant box sizes. All simulations are based on 2LPT initial conditions, generated with `MUSIC` at redshift 49. The power spectrum is measured with the triangular shaped cloud (TSC) mass assignment on a 8192^3 grid.

In the left panel of Fig. 8.4, we illustrate ratios of power spectra from runs with the same mass resolution but different box sizes and particle numbers at redshift zero. The particle mass is kept constant at $M_p \sim 10^{10}\ h^{-1}\mathrm{M}_\odot$, while the box length is increased together with the number of particles. A small box with $L = 128\ h^{-1}$Mpc (blue line) systematically underestimates the power by more than 10 percent. Doubling the box size to $L = 256\ h^{-1}$Mpc box (green line) leads to an overall accuracy of 5 percent (one percent at small scales, $k > 1\ h\,\mathrm{Mpc}^{-1}$). Boxes with length of $L = 512\ h^{-1}$Mpc (red line) and more ($L = 1024\ h^{-1}$Mpc, black reference line) differ by about one percent or less over the entire range of wave numbers. We therefore conclude that the box size of simulations should not be smaller than 500 h^{-1}Mpc in order to eliminate all systematic *nonlinear* finite-volume effects at the required percent precision (see also [70, 135] for similar conclusions). Reducing the sample variance below the percent level over the entire k-range illustrated in Fig. 8.4 would require a much larger box of $L \sim 12\ h^{-1}$Gpc.

In the right panel of Fig. 8.4, we plot power spectra from runs with the same box size ($L = 512\ h^{-1}$Mpc) and different particle numbers, effectively increasing the mass resolution. Simulations with $N = 256$ (blue line), $N = 512$ (green line), and $N = 1024$ (red line) underestimate the power on small scales with respect to the $N = 2048$ reference run (black line). The convergence rate with respect to the percent accuracy requirement is directly proportional to the scale where the simulation shot-noise becomes relevant, i.e. the wave number $k_{\rm sn}$ at which $P_{\rm sn}/P \equiv 0.01$. (illustrated by the dotted vertical lines in Fig. 8.4). The maximum wave number $k_{\rm max}$ warranting percent accuracy is well described by $k_{\rm max} = k_{\rm sn}/3$. For the runs shown in the right panel of Fig. 8.4 with $N = 256$, 512, 1024 (blue, green and red lines) this results in $k_{\rm max} = 0.2$, 1.0, 4.0 $h\,{\rm Mpc}^{-1}$. Previous investigations by Refs. [104, 87] have proposed $k_{\rm max}$ to be half the Nyquist frequency ($k_{\rm Ny} = \pi N/L$) instead. For the same runs this would lead to $k_{\rm max} = 0.78$, 1.58, 3.14 $h\,{\rm Mpc}^{-1}$ which does not exactly reproduce our results[9].

The drop in power of low resolution runs, visible in the right panel of Fig. 8.4, can be explained in terms of analytical considerations: experiments with the halo model show that clusters significantly contribute to the power spectrum, while the presence of small haloes below $10^{11}\ h^{-1}{\rm M}_\odot$ have a negligible effect [170, 45]. Since the simulations with lower resolution (represented by the coloured dots) do not resolve haloes down to masses of $10^{11}\ h^{-1}{\rm M}_\odot$, they underestimate the physical power at small scales. The $N = 2048$ simulation on the other hand, has a particle mass of $M_p \sim 10^9\ h^{-1}{\rm M}_\odot$, resolving $10^{11}\ h^{-1}{\rm M}_\odot$ haloes with ~ 100 particles. Moreover, the convergence rate in the plot suggests that the $N = 2048$ run is one percent accurate until $k \sim 10\ h\,{\rm Mpc}^{-1}$ at redshift zero.

Based on the right-hand-side panel of Fig. 8.4, we can determine a minimal mass resolution required to obtain percent convergence in the matter power spectrum. Since the runs illustrated by the blue, green, and red lines underestimate the power by more than a percent for values above $k = 0.25$, 1, 4 $h\,{\rm Mpc}^{-1}$, we can safely expect the black line to depart from the true answer beyond $k = 10\ h\,{\rm Mpc}^{-1}$. A conservative estimate therefore yields a maximum simulation particle mass of $M_{\rm p} = 10^9\ h^{-1}{\rm M}_\odot$ guaranteeing a percent converged power spectrum at all scales up to $k = 10\ h\,{\rm Mpc}^{-1}$. This requirement can be relaxed to $M_{\rm p} \sim 8 \times 10^{10}\ h^{-1}{\rm M}_\odot$ for wave numbers up to $k = 1\ h\,{\rm Mpc}^{-1}$ (as shown by the green line). Numerical simulations for upcoming survey missions need large boxes of at least 4 h^{-1}Gpc to cover the entire survey volume [98, 115]. This means that at least $N = 16000$ particles per dimension (i.e four trillion in total) are required to reach percent precision for the power spectrum up to $k \sim 10\ h\,{\rm Mpc}^{-1}$.

8.3.3 Best guess for the power spectrum

After investigating the convergence with respect to box size and mass resolution, we present a suite of four simulations with each $N = 2048$ per dimension and decreasing box sizes of $L = 4096$, 2048, 1024, and 512 h^{-1}Mpc. These simulations provide a combined measurement of the power spectrum over the entire range of scales from $k \sim 0.05\ h\,{\rm Mpc}^{-1}$ to $k \sim 10\ h\,{\rm Mpc}^{-1}$.

[9] Assuming a power-law dependence of the power spectrum, $P(k) \propto k^{-\alpha}$, the scaling of $k_{\rm max} = k_{\rm sn}/3$ goes as $k_{\rm max} \propto (N/L)^{3/\alpha}$. For the asymptotic limit of $\alpha = 3$ both approaches - the convergence scale to be tied to the shot noise or to the Nyquist frequency - yield the same scaling with k. For $\alpha < 3$, however, $k_{\rm sn}$ converges somewhat faster, which is in better agreement with our simulations.

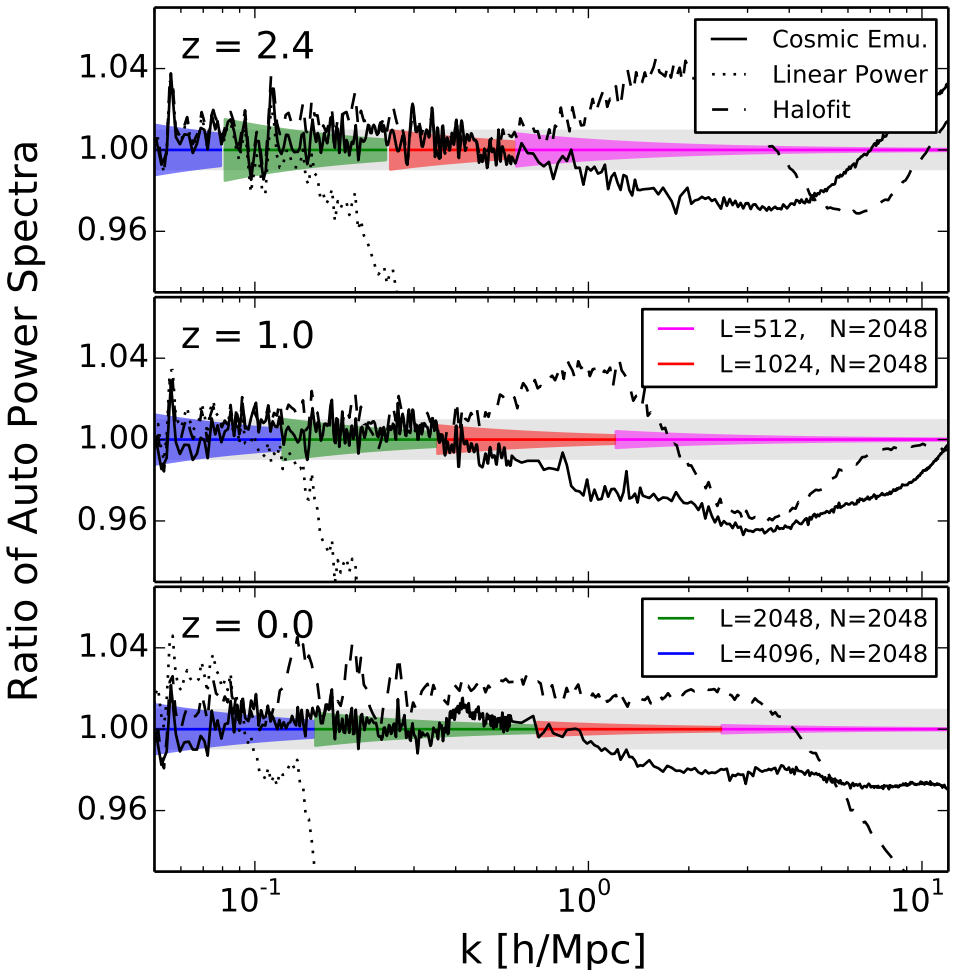


Figure 8.5: Power spectra of the cosmic Franken emulator [86], the revised halofit function [198], and the linear prediction compared to the outcome of simulations with $N = 2048$ particles per dimension and varying box size L=512 h^{-1}Mpc (magenta), L=1024 h^{-1}Mpc (red), L=2048 h^{-1}Mpc (green), L=4096 h^{-1}Mpc (blue). Coloured areas quantify the statistical errors from sample variance and shot-noise, the grey shaded band highlights the range of percent accuracy.

In Fig. 8.5 we use the combined power spectrum from these four simulations as reference line, where the different colours indicate which simulation as been used for which k-range. The colour-shaded areas furthermore indicate the uncertainties due to sample variance and the grey-shaded area delimits the range of percent precision. In Fig. 8.5 the Franken emulator [85, solid black line], the halofit model [198, dashed black line], and the linear theory (dotted black line) are plotted against the reference line from our simulations. The halofit model is a revised version of the Smith et al. [183] fitting scheme, which is physically motivated by the halo model and claims to be 10 percent accurate for $k \leq 1\ h\,\mathrm{Mpc}^{-1}$ between $z = 0$ and $z = 10$. Compared to our simulations the agreement is better than 5 percent over all measured scales and redshifts. The cosmic Franken emulator is an interpolation tool based on a suite of simulations with varying cosmological parameters [87, 86]. The agreement between the emulator and our simulations is about three percent at $z = 0$ and five percent at $z = 1$. This is roughly within the stated accuracy of Heitmann et al. [86]. However, our simulations consistently predict more power than the cosmic emulator at scales around $k \sim 1\ h\,\mathrm{Mpc}^{-1}$ and above, confirming results from Skillman et al. [182] who observe a similar departure from the cosmic emulator. Part of the difference should come from the fact that the emulator was calibrated with `Gadget` runs, while we use `Pkdgrav`, two codes that differ by about three percent at $k > 1\ h\,\mathrm{Mpc}^{-1}$ as illustrated in Fig. 8.1.

8.4 Conclusions

The future of cosmology relies on data from large scale structure surveys. This data can only be fully exploited if we understand gravitational clustering and galaxy formation at high accuracy. The matter power spectrum, as the prime statistical measure, needs to be known within percent precision from linear scales up to $k \sim 10\ h\,\mathrm{Mpc}^{-1}$.

Although cosmological N-body techniques have been developed and constantly improved during the last two decades, obtaining the required accuracy remains a challenge. The entire pipeline from the generation of initial conditions to the analysis of the final data needs to be examined carefully and potential sources of error have to be quantified.

In this paper, we compare power spectra of simulations from the three gravity codes `Ramses`, `Pkdgrav3`, and `Gadget3`. These codes are well established in the community and represent common N-body techniques for cosmological simulations: the particle-mesh technique, the tree method, and a hybrid combination of the two. In a second part, we explore potential error sources from *initial conditions*, *simulation volume*, and *resolution*, investigating effects on the matter power spectrum. These findings are then expressed in terms of a minimal volume and minimal mass resolution requirement to obtain the targeted percent accuracy.

The main results of the paper can be summarised as follows:

1. *Gravity calculation:* The gravity codes `Ramses`, `Pkdgrav3`, and `Gadget3` agree within one percent up to $k = 1\ h\,\mathrm{Mpc}^{-1}$ (over all studied redshifts), and within three percent up to $k = 10\ h\,\mathrm{Mpc}^{-1}$ (below redshift one). Increasing the accuracy of the global time integration is likely to further reduce errors at the largest scales (as suggested by our analysis of the phase spectrum and the bispectrum). Things are likely to be much

more challenging at small scales, as there is no reference solution for the nonlinear power spectrum.

2. *Simulation volume:* A box size larger than $L \sim 0.5\ h^{-1}$Gpc is needed to avoid biases from nonlinear finite-volume effects at the percent level of the power spectrum. Reducing the Gaussian sample variance to a sub-percent level for the k-range above $k = 0.1\ h\,\mathrm{Mpc}^{-1}$ would require an even larger box size of $L \sim 2.5\ h^{-1}$Gpc.

3. *Mass resolution:* A conservative estimate of the maximum particle mass in simulations yields $M_\mathrm{p} = 10^9\ h^{-1}\mathrm{M}_\odot$ for percent accurate power spectra up to $k = 10\ h\,\mathrm{Mpc}^{-1}$. This requirement can be relaxed to $M_\mathrm{p} \sim 8 \times 10^{10}\ h^{-1}\mathrm{M}_\odot$, if only wave numbers up to $k = 1\ h\,\mathrm{Mpc}^{-1}$ are considered. Upcoming surveys, such as `DES`, `LSST`, and `Euclid`, require large simulation volumes of $L \sim 4\ h^{-1}$Gpc or more. As a consequence, numerical simulations need to have at least $N \sim 16000$ particles per dimension (i.e four trillion in total) to reproduce the power spectrum at targeted accuracy.

4. *Initial conditions:* Initial conditions based on the Zel'dovich approximation (ZA) require very high starting redshifts of $z_i = 200$ or above. Such high redshifts are prone to numerical errors, since the size of perturbations are of the order of the numerical accuracy. Initial conditions based on second order Lagrangian perturbation theory (2LPT) are significantly more accurate. They allow late starting redshifts, reducing the run-time of simulations and minimising potential numerical errors in the high redshift regime.

Summarising these results, it is possible to run cosmological simulations with sub-percent errors from volume and mass resolution effects, however, at the price of very high particle numbers. In terms of the gravity calculation, the agreement between codes is good, but not quite at the percent level for the very nonlinear regime.

In the future, it will be crucial to include baryonic effects driven by AGN feedback, as they have been shown to significantly affect the matter power spectrum at nonlinear scales [46, 7]. Quantifying and parametrising the AGN feedback will be one of the main challenges of computational cosmology, and a basic requirement to take full advantage of the upcoming large scale structure observations.

Data Release

All relevant data of the code comparison project, i.e. the IC file, run parameters, and power spectra measurements, can be found at `www.ics.uzh.ch/~aurel/`. We hope that this information will be useful for future comparison and accuracy tests including other N-body codes.

Acknowledgements

This work was initiated within the framework of the Cosmological Simulation Working Group of the Euclid Consortium. AS is supported by the Synergia project "Euclid" from the Swiss National Science Foundation. VS acknowledges support from the Deutsche

Forschungsgemeinschaft (DFG) through Transregio 33, "The Dark Universe". All simulations were run on the *Piz Daint* cluster at the Swiss National Supercomputer Centre (CSCS) under the project allocation s511.

8.5 Variation of code parameters

In the main text, we use standard code parameters from the literature to compare the different gravity codes. This is justified because the exact solution of the power spectrum is not known at nonlinear scales, and a posterior adjustment of code parameters would lead to a false impression of convergence. It is nevertheless important to quantify how the choice of code parameters affects the final results.

In general, code accuracy parameters can either be attributed to the force calculation or the time-stepping. Typical parameters regulating the force accuracy are softening-length and opening-angle for tree-codes (such as `Pkdgrav3`) as well as grid refinement strategy and accuracy of the Poisson solver for adaptive PM codes (such as `Ramses`). Hybrid codes (such as `Gadget3`) usually have an additional parameter regulating the transition scale between the PM and tree regime (PM-grid). The accuracy of time integration, on the other hand, is usually controlled by the adaptive time-stepping criterion which is implemented in a similar way in all three codes.

Past work has shown that for tree codes softening and tree-opening criteria show percent convergence at $k \leq 10\,h\,\mathrm{Mpc}^{-1}$for reasonable parameter choices [159, 186]. The same seems true for the force accuracy parameters of adaptive PM codes which have shown to yield the same precision than generic tree-codes [201]. More significant deviations are reported for the transition parameter between the PM and tree regimes in hybrid codes [186] and for the time-stepping criterion affecting all three codes [159]. In the following, we investigate the effects of both time-stepping and PM-grid transition on the resulting power spectrum.

In order to test the effect of time-stepping, we run simulations with alternative time-stepping criteria for all three codes of the comparison project. For `Ramses` and `Gadget3` we use the global time-stepping mode as an alternative, where *all* particles trajectories are integrated with the smallest time-stepping of the adaptive (default) mode independently of their gravitational acceleration. For `Pkdgrav` we keep the adaptive nature of time-stepping and vary the time-stepping parameter η around the default choice $\eta = 0.15$.

The impact of the time-stepping on the power spectrum is illustrated in the left panel Fig. 8.6. Switching to global time-stepping affects the result at the percent level around $k \sim 10\ h\,\mathrm{Mpc}^{-1}$ for both `Ramses` and `Gadget3`, however, with an inverse general trend (reducing power for `Ramses` and increasing it for `Gadget3`). Varying the η parameter in `Pkdgrav3` around $\eta = 0.1 - 0.2$ also leads to a percent effect on the power spectrum at $k \sim 10\ h\,\mathrm{Mpc}^{-1}$ (bottom panel), with the general trend of increasing power for smaller time-steps. Based on these tests we conclude that the results from the code comparison (i.e., Fig. 8.1) are not sensitive to the time-step criterion as long as reasonable parameter choices are considered.

The effect of the PM-grid transition in `Gadget` is investigated by running simulations with different size of the PM grid around the default choice (where the number of grid points equals the particle number, i.e., PM-grid = N). The resulting power spectra are

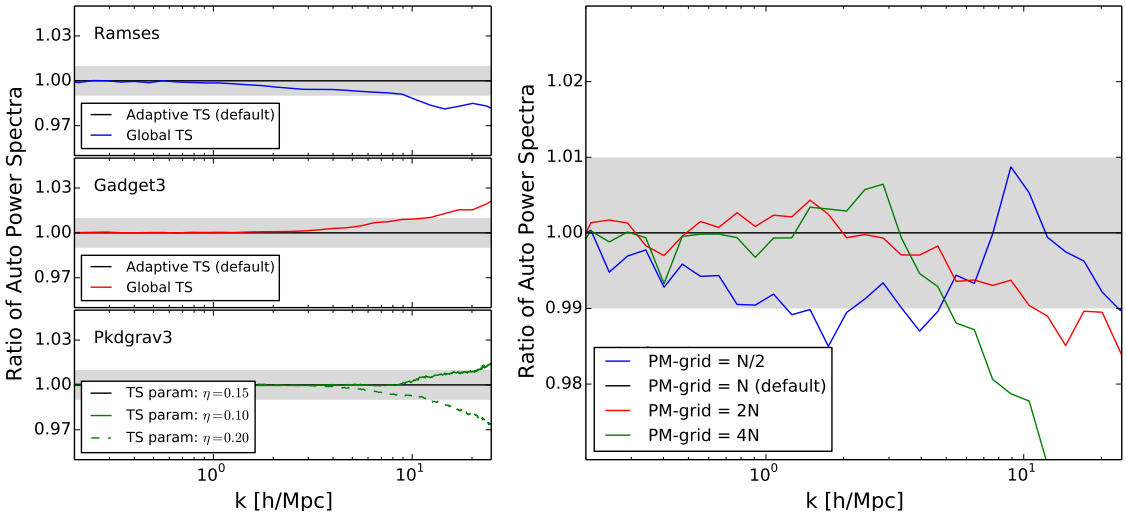


Figure 8.6: *Left:* Different time-stepping strategies and their effects on the auto power spectrum. For `Ramses` (top) and `Gadget3` (centre) we show adaptive (default) and global time-stepping, for `Pkdgrav3` (bottom) we vary the time-stepping parameter η around the default value $\eta = 0.15$. *Right:* Varying grid size of the PM-mesh in `Gadget3` and how this affects the auto power spectrum.

illustrated in right panel of Fig. 8.6, showing differences at the percent level over various scales. The size of the scatter seems significant for precision cosmology and requires further investigation. However, the variation is not large enough to explain the offset between `Gadget` and `Ramses`/`Pkdgrav` in the $z = 0$ panel of Fig. 8.1.

We have shown in this appendix that changing the time-stepping criterion of our codes has a sub-percent effect on the auto power spectrum below $k \sim 10\ h\,\mathrm{Mpc}^{-1}$. The error induced by the PM-tree-transition in `Gadget` is slightly larger but still roughly below one percent. As argued above, other parameters, such as softening and tree opening for tree-codes as well as the accuracy of the Poisson solver for mesh-codes are expected to yield even smaller errors. We therefore conclude that simple tuning of parameters is not enough to bring the different codes into sub-percent agreement. Deeper investigations of the discretisation and integration techniques might be required to achieve this goal.

8.6 Beyond the power spectrum: propagator and bispectrum

In this appendix we consider a different set of statistics from the main text to have a deeper understanding of the differences between the N-body codes and also illustrate the robustness of the results obtained so far. The first is the propagator $G(k)$ which results from the cross-correlation of the initial conditions δ_0 (common to all codes) with the density fluctuations δ at a given redshift,

$$G(k,z) \equiv \frac{\langle \delta(\mathbf{k},z)\,\delta_0(\mathbf{k}')\rangle}{\langle \delta_0(\mathbf{k})\,\delta_0(\mathbf{k}')\rangle}. \tag{8.6}$$

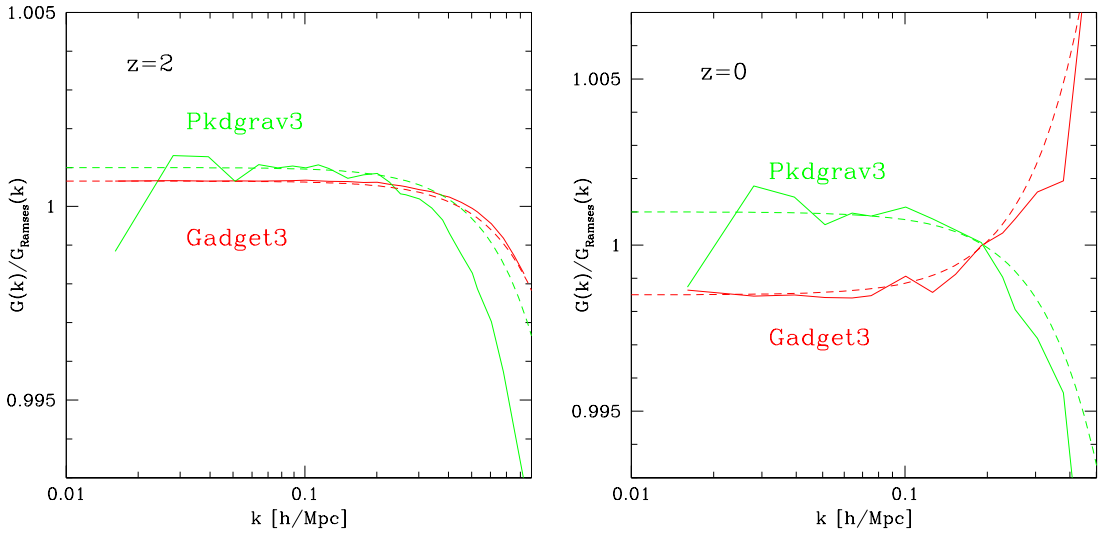


Figure 8.7: *Left:* The solid lines show the ratio of `Gadget3` and `Pkdgrav3` propagators to the `Ramses` propagator at $z = 2$. The dashed lines denote the expectation of these ratios based on RegPT. *Right:* Same as left panel, but at $z = 0$.

This two-point statistic is sensitive to the displacement of particles away from their initial conditions on scales significantly larger than $2\pi/k$, unlike the equal-time power spectrum considered in the main text. At leading order in perturbation theory valid at large scales $G(k)$ agrees with the growth factor, whereas at small scales the propagator drops to zero on scales smaller than the inverse of the rms displacement field at the given redshift [40].

Since we have only one realisation for each N-body code, it is difficult to conclude anything from comparing the propagators to their expected large-scale limit, e.g. computed using RegPT [20]. For example, looking at the first five bins in Fourier space, all three codes agree with the predictions to better than 0.5% at $z = 2$ but there is no clear winner in terms of best agreement as the measurements fluctuate about the theoretical result as k changes. For this reason it is more robust to look at the *ratio* between different codes and explore to what extent the differences between them can be understood.

In Fig. 8.7 we show the ratio of the measured propagators to that of the `Ramses` code, at $z = 2$ (left panel) and $z = 0$ (right panel), shown from the fundamental mode of the simulation box up to scales where the propagator drops (exponentially) to zero. There are three points worth making here. First, the low-k deviations in the propagator ratios are largely consistent (half the value) with those seen in the power spectrum in Fig. 8.1. The second point is that a low-k enhancement (suppression) goes together with a high-k suppression (enhancement) of the propagators. This makes sense as a low-k enhancement corresponds to an overall larger displacement field, which also decreases the propagator at small scales as the cross-correlation between initial and final conditions is suppressed by what is, effectively, slightly more time evolution. Finally, the redshift dependence shows that while the relative behaviour of `Pkdgrav3` and `Ramses` is the same at $z = 0$ and $z = 2$, the `Gadget3` deviations compared to `Ramses` have opposite signs at the outputs considered here.

The figure also shows the expected propagator ratios from RegPT (dashed lines). At $z = 2$ the expectation works very well for the ratio of `Gadget3` to `Ramses` propagators, but

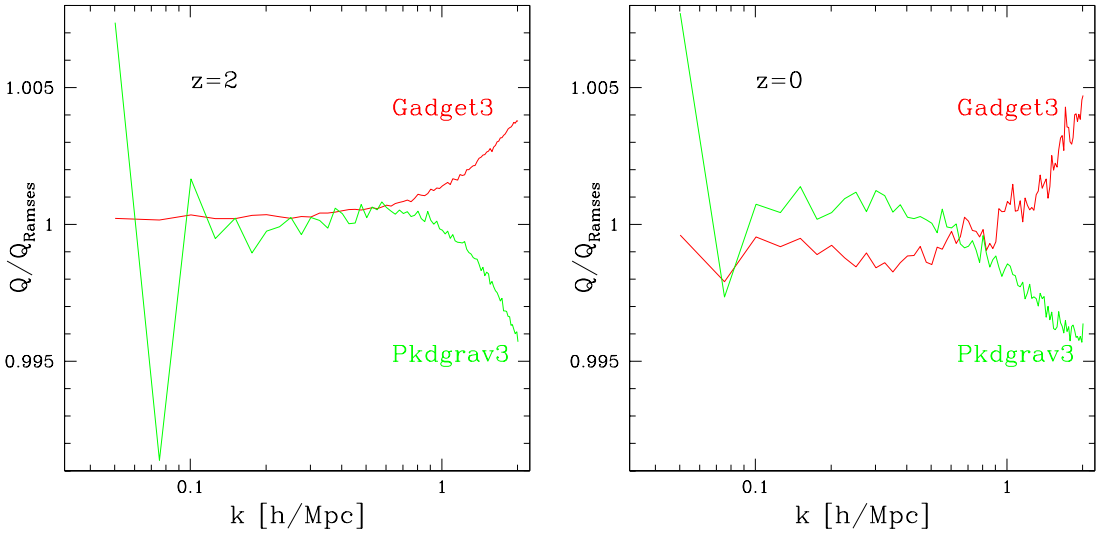


Figure 8.8: *Left:* The ratio of `Gadget3` and `Pkdgrav3` reduced bispectra to the `Ramses` reduced bispectrum at $z = 2$. The reduced bispectrum in each case corresponds to an average over all triangles with maximum wavenumber equal to k. *Right:* Same as left panel, but at $z = 0$.

not so well for the `Pkdgrav3` to `Ramses`, which also shows significant more noise, particularly at the fundamental mode of the box. Clearly, the early (up to $z = 2$) time evolution of `Gadget3` and `Ramses` are consistent with each other except for some small (less than 0.1%) relative displacement, while the evolution of `Pkdgrav3` is not as consistent with `Ramses` (or `Gadget3`) in terms of an overall slight displacement mismatch (as shown by comparison with RegPT and the relative fluctuations. This is perhaps not surprising, as the large-scale forces are computed by the PM method in both `Ramses` and `Gadget3`, while a tree is used in the `Pkdgrav3` case. As the evolution proceeds to $z = 0$, however, the relative evolution of `Gadget3` to `Ramses` drifts and changes sign (at $z = 0$ `Gadget3` is slightly less evolved than `Ramses`) with this relative behavior still fairly well predicted by RegPT. For `Pkdgrav3` the low-k noise remains, but the relative evolution to `Ramses` appears much more consistent than it was at $z = 2$ to what is expected from RegPT.

We now consider the bispectrum. Given the discussion above, we expect that the difference in the bispectra obtained from the different simulations to differ mostly by an overall constant growth factor, and since to leading order in perturbation theory the bispectrum scales as the power spectrum squared, the difference in amplitude between bispectra should be about twice that observed in the power spectrum in Fig. 8.1 (or four times that in the propagator in Fig. 8.7). For this reason, it is convenient to show results for the *reduced bispectrum* Q_{123} defined as,

$$Q_{123} \equiv \frac{B_{123}}{P_1\,P_2 + P_2\,P_3 + P_3\,P_1}, \tag{8.7}$$

where $P_i \equiv P(k_i)$ and $\langle \delta(\mathbf{k}_1)\,\delta(\mathbf{k}_2)\,\delta(\mathbf{k}_3)\rangle = \delta_D(\mathbf{k}_1+\mathbf{k}_2+\mathbf{k}_3)\,B_{123}$ with B_{123} the bispectrum. The reduced bispectrum is, to leading order in perturbation theory, independent of the overall value of the growth factor.

Figure 8.8 shows the results for the reduced bispectra at $z = 0, 2$. We have measured the bispectrum for all triangle shapes from scales of twice the fundamental mode of the box

$k_f \simeq 0.0126\ h\,\mathrm{Mpc}^{-1}$ up to $160\,k_f \simeq 2.01\ h\,\mathrm{Mpc}^{-1}$, using the method discussed in [173]. For simplicity we show the results for the reduced bispectra after averaging all triangles whose maximum side is k (the horizontal axis in Fig. 8.8). We see from this figure that the reduced bispectra are overall in remarkable agreement at the sub-percent level, while the agreement between *bispectra* (not shown) is at the one-percent level at large-scales (as expected from the power spectrum results and the scaling discussed above). In particular, we see that at $z = 2$ the agreement between `Gadget3` and `Ramses` is essentially perfect until $k \sim 0.6\ h\,\mathrm{Mpc}^{-1}$ and then it shoots up by only 0.4% by $k \sim 2\ h\,\mathrm{Mpc}^{-1}$. On the other hand, for `Pkdgrav3` the differences are more noticeable for the low-k modes (as noted for the propagator and power spectrum before). At $z = 0$, for most triangles at low-k the differences remain at the 0.2% level at most, while at high-k the remain below 0.5%. Overall these results are consistent with the picture discussed above, that the evolution of `Gadget3` and `Ramses` are fairly consistent with each other up to an overall small mistmatch in growth factors, while the early time evolution of `Pkdgrav3` differs by a bit more than just an overall scale factor at large scales ($k < 0.1\ h\,\mathrm{Mpc}^{-1}$). In fact, for the sake of clarity Fig. 8.8 starts from $k = 4k_f$, but the ratio of `Pkdgrav3` to `Ramses` reduced bispectra for equilateral triangle of sides $k = 2k_f$ is as large as $\simeq 1.05$ at $z = 0, 2$. Unfortunately there is no reliable way of telling which results, `Pkdgrav3` on the one hand, or `Ramses` and `Gadget3` (which are consistent among themselves), are the correct ones as the cosmic variance is significant at these scales for a single realization of a relatively small simulation box. At small scales all the codes differ in their reduced bispectra at below the one-percent level and the situation is even less clear cut. It seems, however, entirely possible that making the large-scale factors agree better than we have here can move towards making those discrepancies even smaller, as the reduced bispectrum is affected by overall growth in the nonlinear regime. To make this more quantitative, since the *rms* large-scale displacement at $z = 0$ is of order $10\ h\,\mathrm{Mpc}^{-1}$, the error seen on it of order 0.1% (see Fig. 8.7) corresponds to $0.01\ h\,\mathrm{Mpc}^{-1}$ errors on displacements, which of course can give larger than percent corrections to the power spectrum for $k > 1\ h\,\mathrm{Mpc}^{-1}$. In addition, the large-scale enhancement of displacements (or growth factors) which correspond to slightly more evolved configurations do have an enhancement of the small-scale power spectrum, as expected.

Chapter 9

Toward an accurate mass function for precision cosmology

Monthly Notices of the Royal Astronomical Society, Volume 431, Issue 2, p.1866-188

Darren S. Reed[1*], Robert E. Smith[1,2], Doug Potter[1], Aurel Schneider[1]
Joachim Stadel[1] & Ben Moore[1]

[1]Institute for Theoretical Physics, Univ. of Zürich, Winterthurerstr. 190, 8057 Zürich, Switzerland
[2]Argelander-Institute for Astronomy, Auf dem Hügel 71, D-53121 Bonn, Germany

Cosmological surveys aim to use the evolution of the abundance of galaxy clusters to accurately constrain the cosmological model. In the context of ΛCDM, we show that it is possible to achieve the required percent level accuracy in the halo mass function with gravity-only cosmological simulations, and we provide simulation start and run parameter guidelines for doing so. Some previous works have had sufficient statistical precision, but lacked robust verification of absolute accuracy. Convergence tests of the mass function with, for example, simulation start redshift can exhibit false convergence of the mass function due to counteracting errors, potentially misleading one to infer overly optimistic estimations of simulation accuracy. Percent level accuracy is possible if initial condition particle mapping uses second order Lagrangian Perturbation Theory, and if the start epoch is between 10 and 50 expansion factors before the epoch of halo formation of interest. The mass function for halos with fewer than ~ 1000 particles is highly sensitive to simulation parameters and start redshift, implying a practical minimum mass resolution limit due to mass discreteness. The narrow range in converged start redshift suggests that it is not presently possible for a single simulation to capture accurately the cluster mass function while also starting early enough to model accurately the numbers of reionisation era galaxies, whose baryon feedback processes may affect later cluster properties. Ultimately, to fully exploit current and future cosmological surveys will require accurate modeling of baryon physics and observable properties, a formidable challenge for which accurate gravity-only simulations are just an initial step.

9.1 introduction

In the vacuum energy dominated cold dark matter cosmological model [hereafter ΛCDM, 111], large-scale structures form through the amplification of small density fluctuations via gravitational instability. At early times this amplification can be followed using linear perturbation theory of the general relativistic equations of motion for the field. At late times, owing to the nonlinearities in the equations, and after shell-crossing, the dynamics may only be accurately followed using numerical simulations. Overdense regions of the density field, whose dynamics have broken away from the evolution of the background space-time and have reached some state of virial equilibrium are commonly referred to as dark matter halos.

The growth rate of large-scale structures is directly sensitive to the expansion rate of the Universe, and hence the cosmological parameters. One can show theoretically, through the excursion set formalism [154, 27, 179], that the number of halos is also sensitive to cosmological parameters, and importantly for future surveys, the presence of "dark energy" [209, 81, 118, 124, 44, 38]. This forecast cosmological sensitivity has recently been verified through direct testing with N-body simulations [184].

The amount of cosmological information that can be extracted from cluster number counts is limited by our ability to detect signal-to-noise peaks in our observational survey – i.e. associate galaxies to groups, identify groups relative to an X-ray background noise level, etc. The lower that one can push the minimum detectable mass, the more cosmological information can be extracted from the survey. This comes under the proviso that one can accurately calibrate the true–observable mass relationship [118, 125, 164, 123, 143, 8]. The numbers of rare halos are also sensitive to the level of non-Gaussianity in the primordial density field due to its effect upon the tail of extreme density fluctuations [128, 126]. Cluster counts are also sensitive to the total neutrino mass [210, 33, 180, e.g.] Thus, surveys that promise to accurately measure the evolution of the abundance of groups and clusters, also have the potential to help probe fundamental physics. Accurate theoretical predictions for the cluster mass function and its dependence on cosmology, are therefore essential to fully exploit next generation cluster surveys.

Current cosmological constraints from clusters come from: Vikhlinin et al. [207], Vanderlinde et al. [206], Rozo et al. [165], Sehgal et al. [174], Allen, Evrard and Mantz [5] and Ade et al. [2]. Over the next decade there will be a number of large surveys that will aim to strongly constrain the cosmological model through the abundance of clusters: in the X-ray there will be eROSITA [150], with the Sunyaev-Zel'Dovich method there will be Planck, in the optical using the weak lensing method there will be DES, Euclid [115] and LSST. Several authors have estimated the requirements on the theoretical accuracy of the halo mass function to achieve the statistically limited constraints on cosmological parameters. Wu, Zentner and Wechsler [216] point out that, in order to constrain time evolving dark energy models for DES, the theoretical mass function must be known with an accuracy $\lesssim 0.5\%$.

In this paper we address the question: What are the correct numerical parameters

*email: reed@physik.uzh.ch
www.rssd.esa.int/planck
www.darkenergysurvey.org
www.lsst.org/lsst

needed to achieve percent level accuracy in the mass function in a cosmological simulation? Large simulation volumes (whether by a single simulation or by multiple realizations) are able to reduce statistical uncertainties due to finite halo numbers. However, large absolute volumes are needed to reduce systematic and statistical errors associated with poor sampling of large-scale density modes [e.g. 13, 12, 152, 162, 41, 184]. Over the past decade, impressive statistical precision in the halo mass function has been achieved using suites of cosmological simulations [100, 213, 162, 203, 41, 95, 26, 184, 8, 4, 214, and others]. However, statistical precision does not imply accuracy, even when considering gravity-only simulations. Sources of systematic error include finite simulation volume, force resolution, mass resolution and discreteness effects, time-stepping, halo finding, initial condition particle mapping, and start redshift. Recent progress includes Crocce et al. [41] and Bhattacharya et al. [26], who each address many of the systematic uncertainties and determine a halo mass function with an estimated accuracy of $\sim 2\%$ from a suite of large gravity-only cosmological boxes, though their results differ by significantly more than this for halos larger than $\sim 10^{15} h^{-1} M_{\odot}$. Moreover, neither approach has taken into account the full covariance matrix of mass function estimates when deriving best fit parameters [184].

As a first step on the path toward producing an accurate mass function in the observational plane, we limit ourselves to demonstrating how percent level accuracy in gravity-only (i.e. collisionless) simulations (wherein baryons are present but interact only via gravity) can be accomplished. We show how to set up initial conditions so that percent level accuracy can be achieved. We also isolate and test the run parameters that control force resolution (force softening and tree opening angle) and time-step size, allowing us to determine the required values to achieve percent level convergence. Finally, we ask: how many particles do we need to sample the halo mass distribution, in order to obtain a mass function accurate to better than $\lesssim 1\%$.

The paper breaks down as follows: in §9.2 we discuss setting up the initial conditions for the structure formation simulations and the parameters used to run the N-body codes. In §9.3 we describe the suite of N-body simulations performed and halo identification. In §9.4 we present the results for the halo mass function and its convergence with simulation parameters. In §9.5 we explore the variation of the halo mass function with the method for generating the initial conditions. We also make a comparison between the results obtained from two well known N-body codes. In §9.6 we explore the convergence of the matter power spectrum and the 1-point probability density function of matter fluctuations. In §9.7 we discuss the remaining challenges of obtaining better than 1% accurate mass functions from structure formation simulations. In §9.8 we summarize our findings and draw up a set of guidelines for obtaining accurate gravity-only mass functions.

9.2 Simulating structure formation

9.2.1 initial conditions

In order to set up a simulation, we must first select the cosmological model and the probability distribution of the primordial density perturbations. In this study we shall work within the context of the ΛCDM paradigm and assume that the initial density modes are described by a Gaussian random field. The statistics of the field are thus fully specified by

the power spectrum. Hence, a particular realization of the density field in Fourier space may be obtained by drawing a set of uniform random phases and assigning amplitudes drawn from the Rayleigh distribution [61], or through the convolution of white noise with a filter that is related to the power spectrum [23].

A given density field must then be converted into a particle distribution, and several techniques for doing this have been discussed in the literature [e.g. 61, 171, 24, 23, 39]. The traditional approach is to place particles on a uniform lattice, and these are then displaced off the initial points $\mathbf{q}$ using a displacement field $\mathbf{\Psi}(\mathbf{q})$ that encodes all of the statistical properties of the density field. Hence, initial (Lagrangian) and final (Eulerian) positions, $\mathbf{x}$, are related through:

$$\mathbf{x} = \mathbf{q} + \mathbf{\Psi}(\mathbf{q}, \tau) \ , \tag{9.1}$$

where the coordinates $\mathbf{x}$ are a solution to the equation of motion:

$$\frac{d^2\mathbf{x}}{d\tau^2} + \mathcal{H}(\tau)\frac{d\mathbf{x}}{d\tau} = -\nabla\Phi \tag{9.2}$$

where in the above $d\tau = dt/a(t)$ is conformal time, $\mathcal{H} = aH(a)$, and Φ is the peculiar gravitational potential. The solution for Ψ is perturbative, and each order can be found through iteration with solutions of lower order. In terms of the initial density field, and up to second order, the solutions may be written [217, 31, 31, 28, 171]:

$$\mathbf{\Psi}(\mathbf{q}, \tau) = -D_1(a)\mathbf{\nabla}_q\phi^{(1)}(\mathbf{q}) + D_2(a)\mathbf{\nabla}_q\phi^{(2)}(\mathbf{q}), \tag{9.3}$$

where $D_1(a)$ and $D_2(a) \approx -3D_1^2(a)/7$ are the first and second order growth factors suitable for ΛCDM. The potentials $\phi^{(1)}(\mathbf{q})$ and $\phi^{(2)}(\mathbf{q})$ can be found through iteratively solving the Poisson equations:

$$\nabla_q^2\phi^{(1)}(\mathbf{q}) = \delta^{(1)}(\mathbf{q}) \ ; \tag{9.4}$$

$$\nabla_q^2\phi^{(2)}(\mathbf{q}) = \sum_{i>j}^{3}\left\{\phi^{(1)}_{,ii}(\mathbf{q})\phi^{(1)}_{,jj}(\mathbf{q}) - \left[\phi^{(1)}_{,ij}\right]^2\right\} \ , \tag{9.5}$$

where $\phi_{,ij} \equiv \partial^2\phi/\partial q_i\partial q_j$. The linear solutions for $\mathbf{\Psi}$, with $\phi^{(2)} = 0$, yield the traditional Zel'Dovich approximation, which we refer to as 1LPT, and the second order solutions, with $\phi^{(2)}$, we refer to as 2LPT. Scoccimarro [171] gave a detailed prescription for implementing 2LPT displacements in simulations, and we make use of a slightly modified version of the publicly available code `2LPT`. Crocce, Pueblas and Scoccimarro [39] demonstrated that 2LPT reduces numerical "transients" to the level where an accurate representation of the halo mass function may be obtained for relatively late start times, $a_\mathrm{f}/a_\mathrm{i} \approx 10$, where a_i and a_f are the initial and final expansion factors.

As can be seen from Eq. (9.3), in the limit of asymptotically high initial redshift 1LPT and 2LPT become equivalent since $D_2(a_\mathrm{i})/D_2(a_\mathrm{f}) \ll D_1(a_\mathrm{i})/D_1(a_\mathrm{f})$. This has led some to speculate that, provided one takes the initial start redshift to be sufficiently high, then it should not matter whether one uses 1LPT or 2LPT. This issue will be investigated in detail in §9.5.

Several earlier studies have explored the importance of 1LPT versus 2LPT initial conditions: Knebe et al. [108] used `Gadget-2` to show that start redshift and 2LPT versus

http://cosmo.nyu.edu/roman/2LPT

1LPT had little effect on internal halo properties, specifically testing halo concentration, spin parameter, tri-axiality. They also found little dependence on halo mass or the halo mass function. However, their results may understate any numerical issues because they focused on smaller halos of 10^{10}–$10^{13}h^{-1}M_{\odot}$ where the mass function is not very steep, and their statistics were limited due to using low-resolution $N = 256^3$ particles. A more recent study by Jenkins [99], has shown that there is a definite, although weak, dependence of the subhalo mass function inside Milky-Way mass halos on the choice of the initial conditions.

One last issue, concerning the generation of initial Gaussian random density fields, is that some researches have advocated the use of the "Hann filter" [23]. This is an anti-aliasing filter [155], and corresponds to setting the Fourier density modes that are outside the Nyquist sphere of the simulation, $k_{\rm Ny} = \pi N^{1/3}/L$, to vanish by multiplying the transfer function by $W(k) = \cos(\pi k/2k_{\rm Ny})$ for $k < k_{\rm Ny}$ and 0 for $k > k_{\rm Ny}$. The purpose of this is to mitigate some of the anisotropies in the forces due to the cubical lattice. In §9.5 we shall also investigate how the presence of such filtering impacts our goal of an accurate mass function.

Note that for some of the simulations where we test for parameter convergence, we will also make use of a modified version of `Grafic-2` [23]. These two initial condition codes were verified to show identical convergence trends with start redshift.

9.2.2 N-body codes

Once we have obtained an initial condition, we then need to integrate the equations of motion, Eq. (9.2). In this study we shall make use of two standard N-body techniques: `PKDGRAV` V2.2.12 and `Gadget-2`.

`PKDGRAV` is our primary simulation code for this study, an early version of which is described in Stadel [195]. The version of the code we use has been MPI parallelized, and uses the hierarchical tree data structure to organize the individual simulation particles. The gravitational force on each particle is calculated using a multipole expansion with Ewald summation to replicate the simulation cube as an approximation of an infinite periodic universe. The peculiar potential around any given particle is obtained from an hexadecapole expansion of the forces. `PKDGRAV` uses a variable time step criterion that is synchronized for global time-steps. Particle orbits are integrated with the symplectic leapfrog integrator.

`Gadget-2` is a tree-particle-mesh (Tree-PM) code, and full details of which may be found in Springel [192]. The main difference with `PKDGRAV` is that on large scales it uses Fourier based methods to solve for the forces and only uses the tree algorithm to solve for forces on small scales. The solution for the potential is then obtained through an interpolation of the PM and tree forces over the force matching region, and typically this is $\sim 4-5$ mesh cells.

In §9.5 we investigate the mass functions from these different codes and explore the convergence properties with different 1LPT and 2LPT start redshifts. Additionally, we aim to determine the typical values for "generic" run parameters that are required for percent level convergence. In what follows we shall describe parameters that are mainly specific

The "Hann filter" is sometimes (erroneously, according to Wikipedia) referred to as the "Hanning" filter.

to PKDGRAV, but will make reference as to how they apply to Gadget-2 or other codes. Gadget-2 parameters are tested further in Smith et al. [186]. The run parameters that we focus on are:

Tree opening angle Θ: The tree opening angle controls the accuracy of medium and long range forces. It does this by setting the minimum distance between a given particle and tree node below which the tree node will be "opened". Thus the force calculations for a given particle will include contributions from entire nodes and or individual particles. A discussion of how Θ relates to other tree types can be found in Stadel [195].

Softening ϵ: In order to avoid excessively large accelerations and hence excessively short time-steps, the small-scale gravitational interactions must be "softened". This makes sense for simulations of collisonless systems like CDM, where the particles represent large coarse grained elements of the microscopic phase space. PKDGRAV and Gadget-2 both use a softened kernel: gravitational forces approach zero for spatially coinciding particles, and become Newtonian at 2ϵ for PKDGRAV and 2.8ϵ for Gadget-2. PKDGRAV uses the K_3 softening kernal of Dehnen [51] while Gadget-2 uses a spline kernel. The force softening leads to a minimum resolved spatial scale. Throughout, we make use of constant comoving softening.

Time-step η: Each particle is on an adaptive time-step with length proportional to the time-step parameter η. The actual time-step length for each particle is based on the magnitude of its current acceleration $|\mathbf{a}|$, the softening length ϵ, and the time-step parameter η, in accordance with the relation:

$$\mathrm{dt} \geq \eta\sqrt{(\epsilon/|\mathbf{a}|)} \ . \tag{9.6}$$

This technique allows significant computational savings in cosmological simulations when only a small fraction of particles are in dense regions requiring the shortest time-steps.

In summary, we shall investigate how the halo mass function varies with: the initial start redshift; with 1LPT or 2LPT initial conditions; with Nyquist filtering; we shall explore results for two simulation codes; and how variations in Θ, ϵ, η affect our results. Besides these, we shall also explore finite volume effects and mass resolution.

9.3 Simulations

9.3.1 Simulation suite

We have generated a suite of N-body simulations that are designed to explore the accuracy with which we may estimate the halo mass function and its dependence on how we simulate the dark matter (as discussed in §9.2). All of the simulations that we have performed evolve $N = 1024^3$ equal mass dark matter particles. We consider periodic cubes of size $L = 17.625\,h^{-1}$Mpc evolved to $z = 10$, and cubes of size $L = 2048\,h^{-1}$Mpc evolved to $z = 0$. The relative box sizes were chosen so that halos corresponding to $\sim 3\sigma$ fluctuations in the density field are sampled by $N_{\rm h} \sim 1000$ particles at the final output. This corresponds to $M \sim 3.8 \times 10^8 h^{-1} M_\odot$ for the small boxes at $z = 10$, and $M \sim 6.1 \times 10^{14} h^{-1} M_\odot$ at $z = 0$ for the larger boxes. Thus, the final halos in the small box simulations are in an evolutionary state similar to the clusters in the large box simulations at $z = 0$.

Code	IC method	$L\,[h^{-1}\mathrm{Mpc}]$	$\mathrm{m_p}\,[h^{-1}M_\odot]$	z_i	z_f	η	ϵ	Θ	
PKDGRAV	1LPT	17.625	3.88×10^5	30, 49, 100, 200, 400	9.8	0.15	$l_\mathrm{m}/50$	0.55	
PKDGRAV	2LPT	17.625	3.88×10^5	30, 49, 100, 200, 400	9.8	0.15	$l_\mathrm{m}/50$	0.55	
Gadget-2	1LPT	17.625	3.88×10^5	30, 49, 100, 200, 400	9.8	0.2	$l_\mathrm{m}/30$	0.5	
Gadget-2	2LPT	17.625	3.88×10^5	30, 49, 100, 200, 400	9.8	0.2	$l_\mathrm{m}/30$	0.5	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/50$	0.4	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/50$	0.68	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/50$	0.8	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.07	$l_\mathrm{m}/50$	0.55	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.2	$l_\mathrm{m}/50$	0.55	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.25	$l_\mathrm{m}/50$	0.55	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.3	$l_\mathrm{m}/50$	0.55	
PKDGRAV	2LPT	17.625	3.88×10^5	400	9.8	0.6	$l_\mathrm{m}/50$	0.55	
PKDGRAV	1LPT-g	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/50$	0.55	
PKDGRAV	1LPT-g-HF	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/50$	0.55	
PKDGRAV	1LPT-g	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/20$	0.55	
PKDGRAV	1LPT-g	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/10$	0.55	
PKDGRAV	1LPT-g	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/5$	0.55	
PKDGRAV	1LPT-g	17.625	3.88×10^5	400	9.8	0.15	$l_\mathrm{m}/2$	0.55	
PKDGRAV	1LPT	2048	6.05×10^{11}	30, 49, 200	0.0	0.15	$l_\mathrm{m}/50$	$0.55(z > 2)$,$0.7(z < 2)$	
PKDGRAV	2LPT	2048	6.05×10^{11}	30, 49, 100, 200, 400	0.0	0.15	$l_\mathrm{m}/50$	$0.55(z > 2)$,$0.7(z < 2)$	
PKDGRAV	1LPT-g	2048	6.05×10^{11}	400	0.0	0.15	$l_\mathrm{m}/50$	$0.55(z > 2)$,$0.7(z < 2)$	
Gadget-2	2LPT	2048	6.05×10^{11}	30, 200	0.0	0.2	$l_\mathrm{m}/20$	0.5	

Table 9.1: Suite of cosmological simulations. Col. 1: N-body code used. Col. 2: initial condition method, note that we used the `2LPT` code throughout, except where there is a *-g*, which denotes the use of `Grafic-2`; *-HF* denotes use of a Hann filter. Col. 3: box size. Col. 4: particle mass. Col. 5: initial condition start redshifts that have been simulated. Col. 6: final redshift. Col. 7: time-stepping parameter. Col. 8: Force softening, ϵ, in units of the mean inter-particle spacing l_m. Col. 9: tree opening angle (`ErrTolTheta` for `Gadget-2` runs). The additional `Gadget-2` parameters were set to: `ErrTolIntAccuracy`=0.02, `MaxRMSDisplacementFac`=0.2, `MaxSizeTimestep`=0.02, `MinSizeTimestep`=0.000, `ErrTolForceAcc`=0.005, `TreeDomainUpdateFrequency`=0.05, `PMGRID=1024`.

Although the $L = 17.625\,h^{-1}$Mpc box is very small, the effects of finite volume on our study are attenuated because we examine halos early, at $z = 10$, when the typical halo mass-scale is still much smaller than the total simulation mass. There is no need to apply a finite volume correction as in e.g. Reed et al. [162] to these simulations because each of our convergence series utilizes identical initial conditions and particle displacements. Finite volume effects, to the extent that they can be accounted for with a simple linear correction technique, are thus identical within each convergence series.

The cosmological parameters that we adopted for the small box runs were consistent with WMAP5 [110]: $\Omega_m = 0.274$, $\Omega_\Lambda = 0.726$, $\Omega_b = 0.046$, $h = 0.705$, $n_s = 0.96$, $\sigma_8 = 0.812$, where these parameters are the density parameters in matter, vacuum energy, and baryons; the dimensionless Hubble parameter; the primordial power spectral index; and the variance of the density fluctuations on scales of $R = 8\,h^{-1}$Mpc. The transfer function that we used to create the initial conditions was produced using the prescription of Eisenstein and Hu [63]. For the large box runs, the cosmological parameters we adopted were consistent with WMAP7 [111]: $\Omega_m = 0.2726$; $\Omega_\Lambda = 0.7274$, $\Omega_b = 0.046$, $h = 0.704$, $n_s = 0.963$, $\sigma_8 = 0.809$. The transfer function for these runs was computed using CAMB [116]. Full details for the entire suite of simulations are given in Table 9.1.

9.3.2 Halo identification

We identify dark matter halos using the *friends-of-friends* (FoF) algorithm [47]. This links together all particles that are separated by less than the linking length b, where b is expressed in units of the mean inter-particle separation. Throughout, we use $b = 0.2$, and we use the particular implementation of the algorithm internal to `PKDGRAV`; a similar implementation is provided by the code `fof`. The estimated iso-density contour that the FoF recovers is $\delta = \delta\rho/\bar{\rho} = n_c b^{-3} - 1 \approx 81.62$ [139].

In the literature there is a wide range of alternate approaches to the identification of dark matter halos: e.g. the *spherical over-density* (SO) algorithm [113, 203]; 6-D phase space algorithms [18]; and for a review of methods see Knebe et al. [107]. Rather than explore all of these different methods here, we shall work under the assumption that: an accurate FoF mass function implies an accurate mass function for all other algorithms designed to select *approximately* virialised objects. Anecdotal support for this is provided by McBride et al. [130], who found similar convergence behavior with simulation set-up for the FoF ($b = 0.2$) and SO 200 mass functions. We shall reserve the task of establishing the veracity of this assumption for future work.

A number of systematic errors have been noted for halos obtained with the FoF algorithm. Firstly, the recovered halo masses are systematically overestimated with respect to the "true" halo mass [213, 121, 205, 139]. This owes to the fact that the true halo mass distribution is sampled by a finite number of particles. This mass overestimation has been quantified for spherical mock halos by Lukić et al. [121] and more recently by More et al. [139]. Secondly, FoF halos also experience "bridging", which may occur when two distinct halos undergo a close encounter. This systematic effect: links unvirialised systems, it acts to reduce the overall number of halos found, and it predominantly occurs for the highest mass objects and is stronger at higher redshifts [121].

---

www-hpcc.astro.washington.edu/tools/fof.html

Warren et al. [213] and Bhattacharya et al. [26] have provided empirical corrections, determined from cosmological simulations, for the systematic FoF errors. However, these empirical models are mass and redshift independent, which may make them insufficient for our goal of achieving a $\sim 1\%$ accurate mass function. We would expect the FoF errors to include dependencies on the specific distribution of mass within halos, which depends on both mass and redshift. Nevertheless, we assert that the FoF mass overestimation should not affect our convergence tests because they are all performed at the same mass resolution. For these reasons, we use the raw FoF masses, and remark that this will affect our ability to recover an "unbiased", FoF mass function. However, this is sufficient to our purposes of quantifying relative differences in the mass function of different simulations.

9.4 Mass Function Convergence I: Simulation Parameters

In this section, we show convergence of the FoF halo mass function with varying simulation run and set-up parameters. We estimate the number of halos per logarithmic mass interval, $dn(M)/d\log_{10}M$, using a Gaussian kernel in log mass. The Gaussian kernel is convenient for diagnostics because it avoids the 'saw-tooth' pattern that emerges in a binned mass function as a result of the discretization of halo masses, particularly at low masses where the simulation particle mass is a significant fraction of the mass-width of a bin. The number of halos with mass M_k is estimated by the sum:

$$N_k = \frac{\sum_i M_{hi} f_g}{\sum_i M_{hi}} \ , \tag{9.7}$$

where f_g is a Gaussian kernel (in $log_{10}M$) of width $h = 0.0625$, chosen to minimize Poisson fluctuations without introducing systematic error to the mass function. To minimize computational cost, we truncate the kernel beyond a range of $\pm a = 3h$. The number of halos per logarithmic mass interval per unit volume is:

$$\frac{dn(M)}{d\log_{10}M} = \frac{N_k}{Vh \ \mathrm{erf}(a/\sqrt{2})} \tag{9.8}$$

where V is the simulation volume. The kernel mass M_k is estimated by the following:

$$M_k = \frac{\sum_i N_{hi} M_{hi} f_g}{\sum_i N_{hi}} \tag{9.9}$$

analogous to using the average halo mass in a bin.

Poisson errors can be estimated from the Gaussian kernel halo numbers N_h, through use of the formula ([83]; utilized for the halo mass function in [120]):

$$\sigma_\pm = \sqrt{N_h + \frac{1}{4}} \pm \frac{1}{2} \ . \tag{9.10}$$

In what follows, we will show results for all halos with more than 8 particles per halo. This is done for purely diagnostic purposes, since the poorly resolved halos are strongly affected by particle shot-noise errors.

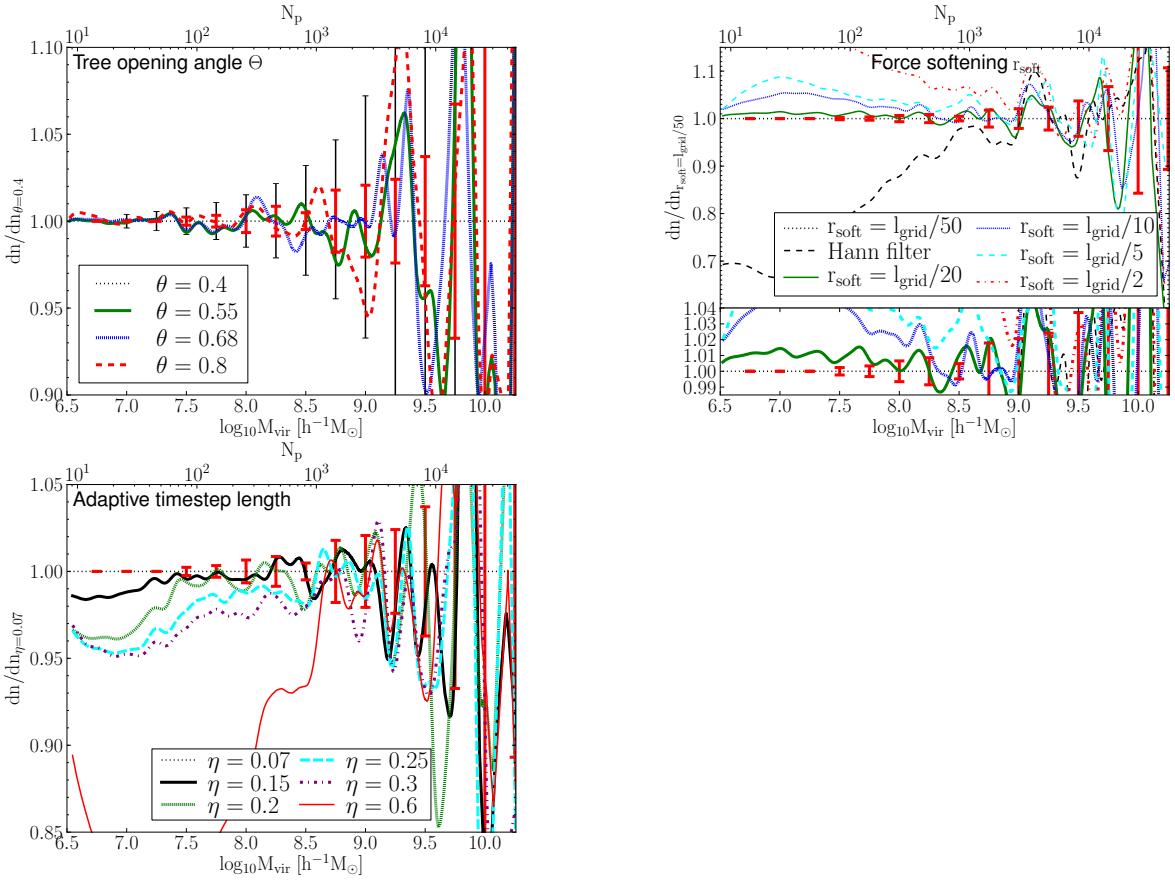


Figure 9.1: Variation of the halo mass function with simulation parameters, relative to the mass function obtained from our fiducial simulation, as a function of FoF halo mass. All panels show the results from the $L = 17.625\,h^{-1}$Mpc, PKDGRAV simulations at $z = 10$. All runs used the same initial realization of the density field. Long thin error bars show the Poisson errors estimated from the number of halos. Short thick red error bars are estimated from the scatter of the mass functions of sub-sampled versions of the simulations. The top, middle and bottom panels show the results of variations in tree opening angle Θ, force softening ϵ, and adaptive time-step parameter η, respectively. The black dashed line in the middle panel shows the large (and assumedly undesirable) impact of application of the Hann anti-aliasing filter to the initial density field, on the halo mass function. Percent level convergence is apparent for each run parameter for halos larger than ~ 100 particles.

9.4.1 Tree opening angle Θ

The top panel of Figure 9.1 presents the results from our convergence study of the tree-opening angle parameter Θ, used in the code `PKDGRAV`. These results are for the case of the $z = 10$, $L = 17.625\ h^{-1}$Mpc runs. The figure clearly shows that, for halos with masses $M \leq 10^9 h^{-1} M_\odot$, the runs with $\Theta \leq 0.68$ are converged at the sub-percent level. For halos with masses $M \gtrsim 10^9 h^{-1} M_\odot$ ($N_p \gtrsim 3000$), the figure shows that scatter in the mass function begins to dominate our results. This implies that systematic errors at greater than the percent level cannot be ruled out for more massive halos, which may be most affected by tree-related criteria. In this case, the $\Theta = 0.8$ run deviates modestly from the other runs. Larger values of Θ have been shown to cause significant direct force errors [195].

The increased scatter in the mass function for halos with $M \geq 10^9 h^{-1} M_\odot$ may seem somewhat surprising, given that all of our simulations had the same initial realization of the density field so that there is no "sample variance" between them. However, even the most accurate particle simulation is still essentially a Monte Carlo representation of a mass distribution. This means that the mass of each halo has a significant uncertainty, which can at best be equal to the square root of the number of its particles. The scatter in the mass function arises from a convolution of the true halo number counts with the scatter associated with simulating, sampling, and measuring the masses of halos (see discussion in §9.3.2). Hence it is non-trivial to determine the true uncertainty. Fig. 9.1 shows the expected Poisson errors (long thin error bars), and one can see that differences between well-converged runs are at sub-Poisson levels.

For a better estimate of the uncertainty, we create randomly subsampled 1-in-8 particle volumes from the full simulation snapshot and measure the $1 - \sigma$ scatter between the FoF mass functions of the subsampled volumes. In Fig. 9.1, the results of this exercise are denoted by the short, thick, red error bars. This scatter tends to be an overestimate of the true uncertainty, since the scatter in the FoF mass will be larger in the randomly sampled volume due to the smaller numbers of particles per halo. For the most massive halos, this overestimation may become worse due to the increased effects of halo bridging (or unbridging), which has a large effect on halo masses and thus on the inferred mass function. Taking these issues into account, we estimate that we are sensitive to percent level systematic shifts in the FoF mass function for halos with less than ~ 3000 particles.

9.4.2 Force softening ϵ

The central panel of Figure 9.1 presents the results from our convergence study of the force softening parameter ϵ. As can be seen, the commonly used softening value of $\epsilon = l_{\rm m}/20$ is converged at near the percent level over all masses ($l_{\rm m} = L/N^{1/3}$). We also see that the low-mass end of the mass function is very sensitive to the choice of ϵ. Halos with $N \gtrsim 1000$ particles appear only weakly dependent on ϵ, provided $\epsilon \lesssim l_{\rm m}/10$ and within our statistical limitations.

For the cases where $\epsilon \gtrsim l_{\rm m}/10$, forces do not become Newtonian until beyond the FoF linking length of $l_{\rm m}/5$. Interestingly, for these large softening lengths, we find that the halo abundances at a fixed mass are increased. One possible explanation for this effect is that the lower central densities of the heavily softened halo profiles [138, 68, 153, 160, 203, e.g.] lead to higher densities near the outer edges of halos. This increased outer-shell

density, implies larger FoF masses as more particles are linked to the outer layers [26]; (see §9.6.2). For the smallest halos with few particles, the virial radii and softening are of similar order. The minimum resolved halo mass thus increases as softening increases [120, e.g.]. These issues have implications for a common running mode for cosmological simulations where computational speed-up at high redshifts is achieved by using a "physical softening", wherein the comoving softening length is scaled by the expansion factor, with a typical maximum of $\mathrm{f}_{\mathrm{soft\ max}} \sim 10\epsilon$. Effectively, in this mode, the softening is frozen in physical coordinates at scale factor $\mathrm{a} = 1/\mathrm{f}_{\mathrm{soft\ max}}$. Our tests suggest that allowing a comoving softening larger than $l_{\mathrm{m}}/20$ at high redshifts leads to significant numerical error for the early forming halos, which is likely to affect high density regions at late times.

Finally, the black dashed line in the central panel of Fig. 9.1 shows the results of applying the Hann anti-aliasing filter. Whilst it may help to correct errors due to the an-isotropic lattice structure, it introduces a $\sim 30\%$ suppression in the number of lower-mass halos relative to the unfiltered initial conditions run and relative to the expected nearly power-law behavior of the mass function predicted from theory [27]. Hann filtered and unfiltered runs only agree at the percent level for halos with $N \gtrsim 3000$ particles.

9.4.3 Time-step size η

The bottom panel of Figure 9.1 presents our study of how variations in the adaptive time stepping parameter η affects the estimated mass functions. The results demonstrate that, for halos with $M \lesssim 10^9 h^{-1} M_\odot$ (3000 particles), an increase in η leads to a decrease in the abundance of halos. We find that percent level convergence in the mass function can be achieved with $\eta \approx 0.15$ for all halo masses, or $\eta \approx 0.2$ for a 100 particle minimum mass. This value is consistent with the value of $\eta = 0.2$ found by Power et al. [153], who examined the convergence behaviour of the density profiles of dark matter halos. This similar converged time-step size is not surprising given that low-redshift halo centers consist of some of the earliest material to be bound into halos [55]. For halos with $N \gtrsim 1000$ particles, the mass function converges with a much longer time-step corresponding to $\eta = 0.6$.

Finally, we note that the value $\eta = 0.2$ for `PKDGRAV` is similar to the default size of the adaptive time stepping in `Gadget-2`: the parameter `ErrTolIntAccuracy` $= \eta^2/2$ has a default setting of 0.025, which corresponds to $\eta = 0.22$.

9.5 Mass Function convergence II: Initial conditions

9.5.1 Results: Small boxes at z=10

Figure 9.2 compares the behaviour of the 1LPT and 2LPT initial conditions, as a function of the start redshift z_{i}, for the small-box simulations at $z = 10$ evolved with `PKDGRAV`. The top and middle panels show the ratios of the halo mass functions for different start redshifts with the halo mass function obtained from the start redshift $z_{\mathrm{i}} = 800$, for 1LPT and 2LPT, respectively. The bottom panel presents the ratio of the 1LPT and 2LPT mass functions for simulations with the same start redshift. We see that both the 1LPT and 2LPT initial conditions converge to yield the same mass function as start redshift increases. However,

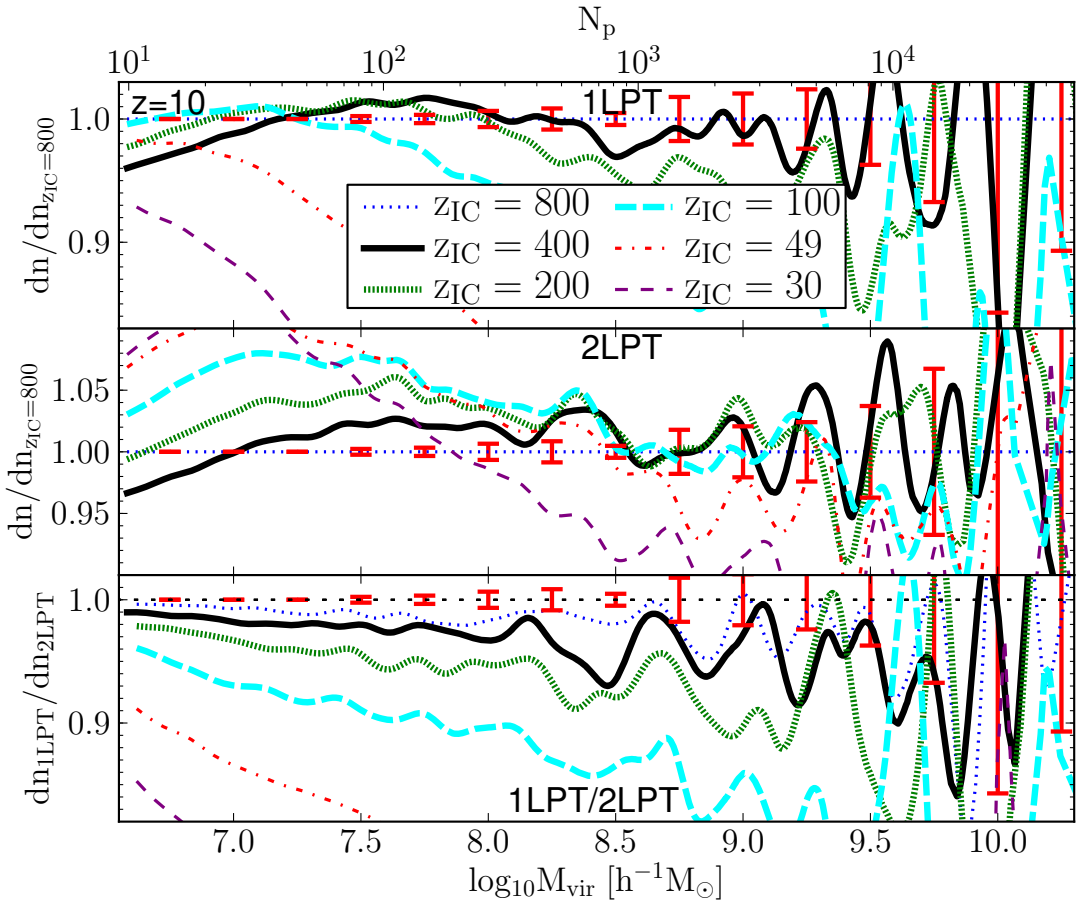


Figure 9.2: Variation in the halo mass function as a function of start redshift, for 1LPT and 2LPT initial conditions using the N-body code PKDGRAV. *Top panel*: results for 1LPT initial conditions. *Middle panel*: results for 2LPT initial conditions. *Bottom panel*: the ratio of the 1LPT and 2LPT mass functions approaches unity at high initial start redshifts. All panels show results of the $L = 17.625\,h^{-1}$Mpc box at $z = 10.0$. The 1LPT series displays 'false convergence' for ~ 100 particle halos. Percent level convergence is met for 2LPT for $N_P \gtrsim 1000$ particles to $z_\mathrm{i} = 400$ or $a_\mathrm{f}/a_\mathrm{i} \approx 40$.

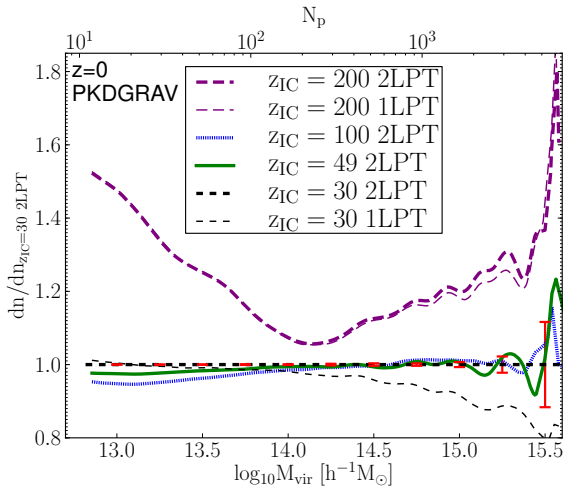


Figure 9.3: Mass functions from the simulations started with 1LPT and 2LPT initial conditions in the $L = 2\,h^{-1}$Gpc boxes at $z = 0$, relative to the simulation started at $z_\mathrm{i} = 30$ with 2LPT initial conditions. All simulations here were run with `PKDGRAV`. Percent level convergence with 2LPT is found between $z_\mathrm{i} = 30$ and $z_\mathrm{i} = 49$ runs at ~ 1000 particles. Extremely early starts ($a_\mathrm{f}/a_\mathrm{i} \gtrsim 100$) lead to serious errors.

the convergence properties of the 1LPT runs are very slow, whereas the 2LPT runs appear to converge much faster.

For the case of 2LPT, we notice that percent level convergence can be achieved for halos with at least ~ 1000 particles and in simulations that have undergone $a_\mathrm{f}/a_\mathrm{i} \gtrsim 10$ expansions. For 1LPT, the $a_\mathrm{f}/a_\mathrm{i} = 80$ run ($z_\mathrm{i} = 800$) is barely converged down to $N \sim 1000$. For halos, with $N \lesssim 1000$ particles, even the 2LPT mass function is poorly converged with z_i for all expansion factors tested. The abundances of small halos appear to diminish as start redshift is increased; this is apparent in both the 2LPT and 1LPT tests. This suggests that $N \sim 1000$ particles represents a minimum halo mass for stability to z_i.

A curious coincidental feature of the 1LPT initial condition series is that small halos appear to be converged at the $\sim 2\%$ level by $z_\mathrm{i} = 200$ (except for the largest masses) and nearly at the $\sim 1\%$ level by $z_\mathrm{i} = 400$ (except for the smallest masses). The more accurate 2LPT start redshift series confirms that this 1LPT convergence is an illusion. With later start redshift, the increased number of halos due to more accuracy in the 1LPT initial conditions is offset by independent errors that act to decrease the number of halos, resulting in *false convergence*. This highlights the fact that convergence is a necessary but not sufficient condition to guarantee accuracy. Some previous studies that appeared to show good mass function convergence with early enough 1LPT initialization, such as Reed et al. [161] and Reed et al. [162] (Fig. A1), among others, likely also suffered from this false convergence. Our larger particle numbers and corresponding better statistical uncertainty, combined with 2LPT comparisons, allow us to make more robust conclusions.

9.5.2 Results: Large boxes at z=0

In order to confirm that the convergence behavior of the small box simulations at $z = 10$ can be applied to the cluster regime at lower redshifts, we present the results from tests run to $z = 0$ in the $L = 2\,h^{-1}$Gpc boxes.

Figure 9.3 shows the results from the 1LPT and 2LPT $z_{\rm i}$ convergence simulations run with the code PKDGRAV. As for the case of the small-box simulations at $z = 10$, we see that low-mass halos are missing with high start redshifts $z \gtrsim 49$; the $z = 0$ "pivot" mass-scale, below which convergence is poor, is somewhat smaller at $N \sim 300$ particles. For larger halos, the 2LPT mass function is well-converged so long as $z_{\rm i} \lesssim 49$; the $z_{\rm i} = 100$ curve deviates from the $z_{\rm i} = 30$ curve at just over the percent level at this mass scale, so is marginally statistically consistent at the percent level. A striking feature here is that when a high enough start redshift is used that Zel'Dovich and 2LPT initial conditions are converged, $z_{\rm i} = 200$, serious errors are present in the $z = 0$ mass function. A likely explanation is that with such a high $z_{\rm i}$, cosmological perturbation amplitudes becomes smaller than the effective amplitude of spurious numerical perturbations. In this $z_{\rm i} = 200$ run, spurious halos begin to form at very early times, initially dominated by 8 particle structures; visual inspection reveals that the spurious halos are aligned with the initial grid of particles. The effects of these early spurious halos lead to the over-abundance of halos at $z = 0$. This underscores the point that 2LPT initial conditions are preferable because they allow one to start at lower redshift where numerical errors are more controllable.

We note that pure particle mesh codes may perform better with high redshift starts because the PM technique is well-suited to following low amplitude linear perturbations. A tree code (and also a tree-PM code), however, is subject to force errors that may accumulate over time, even if they are small, because these errors are correlated with the tree structure. Tree code force errors could thus seed spurious structures that dominate over real cosmological perturbations when start redshift is very high (and initial cosmological perturbations are very low). Further, the accumulated errors would be expected to worsen if time-step length is decreased. The PM code, although it may perform better at early times, is limited in spatial resolution by the mesh size, which is typically much larger than the softening scale of a tree (or tree-PM) code, making it non-ideal for modeling the internal properties of halos.

9.5.3 Transformation to universality

It has been noted that when halo masses are translated into equivalent 'peak-height', i.e., where $M \to \nu(M, a)$, then the mass function takes on a 'universal form' [179]. The peak height is defined through the relation, $\nu = \delta_c/\sigma(M, z)$, where $\delta_c = 1.686$ is the present day linearly extrapolated over-density threshold for collapse in the spherical collapse model, and where $\sigma(M, z)$ is the variance of matter fluctuations on mass scale $M = 4\pi R^3\bar{\rho}/3$. Note also that owing to the fact that $\sigma(M, a) \propto D(a)$, $\nu \propto D^{-1}(a)$, where $D(a)$ is the linear theory growth factor.

In Figure 9.4 we present the ratio of the 1LPT with the the 2LPT mass functions for the small box at $z = 10$ and the big box at $z = 0$ as a function of ν. We see that the 1LPT mass function error appears to be relatively independent of mass and redshift for equal values of ν. This universal behavior is expected in Press-Schechter formalism wherein mean halo formation time depends only upon ν, and δ_c is independent of redshift. We present a fit to the ratio of the 1LPT to 2LPT mass function for our range of data $1 \lesssim \nu \lesssim 5$,

$$\mathrm{dn_{1LPT}}/\mathrm{dn_{2LPT}} = \mathrm{e}^{-0.12\frac{\mathrm{a_i}}{\mathrm{a_f}}\nu^{2.5}}. \tag{9.11}$$

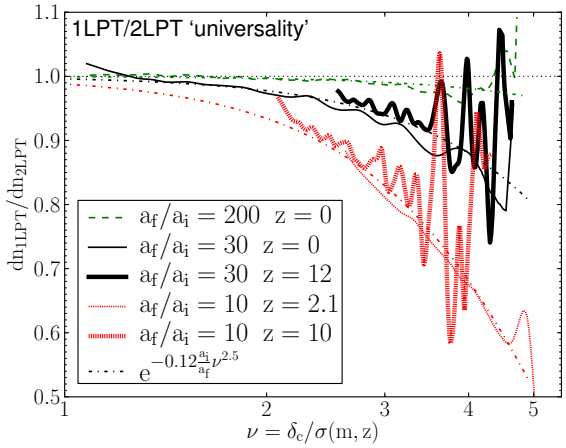


Figure 9.4: Ratio of 1LPT mass functions with those obtained from 2LPT initial conditions as a function of the equivalent halo peak height ν. The thick solid black and thick hatched red line show results from the $L = 17.625\,h^{-1}$Mpc boxes at $\mathrm{z} \geq 10$, and the thin green dashed, solid black and hatched red show results from the $L = 2\,h^{-1}$Gpc boxes. All results here are for PKDGRAV. The ratio of 1LPT/2LPT mass functions is 'universal' in that it depends mainly upon $a_\mathrm{f}/a_\mathrm{i}$ and ν, independent of the specific halo mass range and redshift, and fit by the dot-dashed line.

Percent level convergence between 1LPT and 2LPT initial conditions at scales relevant for the cluster mass function is not achieved until extremely early starts at $a_\mathrm{f}/a_\mathrm{i} \sim 200$. However, such early start redshifts lead to very small initial density perturbations, which through the relative increase of numerical errors, preclude the codes that we tested from being able to accurately follow the growth of structure.

9.5.4 Comparison of PKDGRAV with Gadget-2

The quest for obtaining mass function predictions accurate to within one percent requires different N-body simulation codes to provide consistent results at this level. Of course, if results disagree at $\gtrsim 1\%$ for any two codes, then one would need at least three independent N-body codes to break the degeneracy and so decide which results were correct. Having said that, we now compare the initial condition convergence series of simulations obtained using PKDGRAV with results obtained from the widely used N-body code Gadget-2. Note that we have made no attempt to find 'optimal' parameters for Gadget-2, but instead we have adopted some rather generic choices for these. The full list of simulations that we have performed with Gadget-2, including the exact choices for run parameters, are presented in Table 9.1.

In Figure 9.5 we compare the 1LPT and 2LPT initial conditions as a function of z_i, but this time using Gadget-2, plotted here down to the limit $N = 20$ particles. We find that the results exhibit almost identical behaviour to those obtained from the PKDGRAV runs. The small difference is that the suppression of the mass function at low masses with increasing start redshift, apparent in 2LPT runs for halos with fewer than $N \sim 1000$ particles, is slightly milder with Gadget-2. This appears to enhance the effect of "false convergence" of the 1LPT Gadget-2 simulations.

Figure 9.6 shows the ratio of the mass functions obtained from `Gadget-2` with those obtained from `PKDGRAV`. Note that we used identical initial conditions in all cases. We find that `Gadget-2` systematically produces up to a 10% higher mass function for low-mass halos ($N \lesssim 200$) than `PKDGRAV`. This excess abundance slightly increases with increasing start redshift. We note that the differing mass functions could be a result of differing halo structure, which could lead to systematic differences in FoF masses, and does not necessarily mean the codes are truly producing different numbers of halos. We also note that `Gadget-2` appears to have several percent fewer halos at $N \sim 1000$ particles, for the highest start redshifts.

Figure 9.7 compares 2LPT mass functions from `Gadget-2` and `PKDGRAV` in the large box at $z = 0$. This figure shows that when the lower redshift start is used, $a_{\rm f}/a_{\rm i} = 30$, the two codes generally agree within 2%. However, when high redshift starts are used, $a_{\rm f}/a_{\rm i} \sim 200$, the codes diverge from each other and from the true answer – recall Figure 9.3 where we showed that the lower redshift start is converged in `PKDGRAV` runs.

This code comparison shows that there is a weak systematic shift with start redshift between the codes. However, the convergence behavior of `Gadget-2` with start redshift and of 1LPT versus 2LPT still provides useful verification of the `PKDGRAV` initial condition tests. Ultimately, a robust comparison of absolute accuracy between codes would require that run parameters for each code are run at self-converged values. The code differences are consistent with the level of agreement found between these same codes in Heitmann et al. [85] when considering our improved statistics and resolution. Further investigation is warranted to reveal whether the differences between the two codes is caused by inherent differences between the TreePM and the pure tree method or whether they are instead due to the use of non-ideal run parameters in `Gadget-2`. This is beyond the scope of this paper and we shall reserve a wider study for future work.

9.6 convergence of other properties

In this section we consider the sensitivity of other dark matter statistics: to the adopted simulation parameters; to whether we employ 1LPT or 2LPT initial conditions; and to the adopted start redshift. We shall restrict our exploration to the mass power spectrum and the 1-point Probability Distribution Function (PDF) of dark matter density, as additional diagnostics for determining simulation accuracy.

9.6.1 Mass power spectra

For a finite cubical patch of the Universe, the matter power spectrum is defined to be:

$$\left\langle \delta_{\mathbf{k}_1} \delta^*_{\mathbf{k}_2} \right\rangle \equiv P(|k_1|)\delta^K_{\mathbf{k_1},\mathbf{k_2}}/V_\mu \tag{9.12}$$

where V_μ is the volume of the patch and $\delta_{\mathbf{k}}$ is the discrete Fourier series expansion of the density field. For equal mass dark matter particles, the discrete representation of the

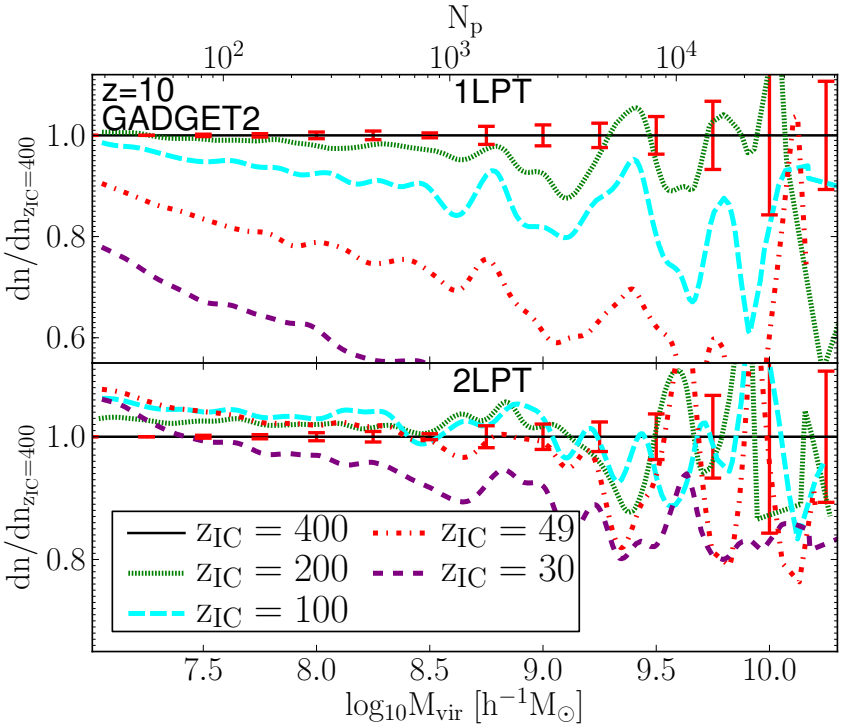


Figure 9.5: Variation in the halo mass function as a function of start redshift, for 1LPT and 2LPT initial conditions using the N-body code `GADGET-2`. *Top panel*: results for simulations started with 1LPT initial conditions. *Bottom panel*: results for simulations started with 2LPT initial conditions. All panels are for the $L = 17.625\,h^{-1}$Mpc box at $z = 10.0$. Convergence of `GADGET-2` runs with initial conditions are very similar that found for `PKDGRAV` (Fig. 9.2).

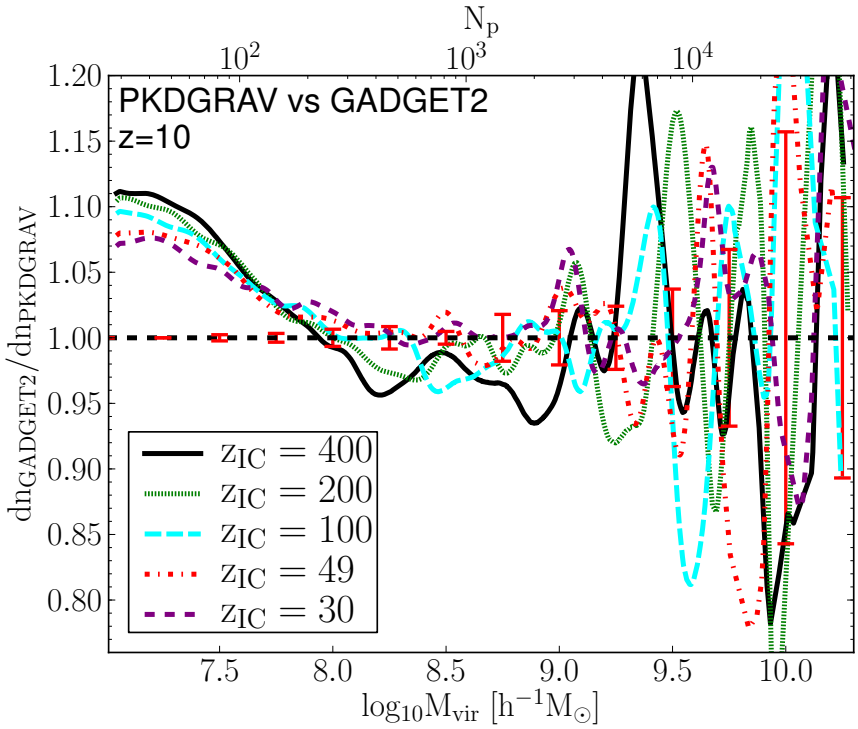


Figure 9.6: $z = 10$. 2LPT. Mass functions from `PKDGRAV` and compared with `Gadget-2`. Some differences are present at small scales but the relative offset is independent of start redshift, providing some confirmation of our initial condition convergence criteria. We have verified the `PKDGRAV`, but not the `Gadget-2` run parameters for percent-level self-convergence.

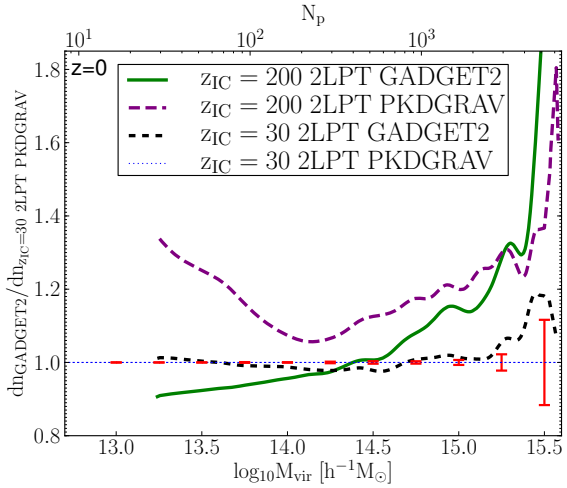


Figure 9.7: Mass functions from `Gadget-2` compared with `PKDGRAV` in the $L = 2\,h^{-1}$Gpc box. The agreement with `Gadget-2` using relatively standard run parameters is at the couple percent level with $z_i = 30$. Extremely early starts ($a_f/a_i \gtrsim 100$) lead to serious errors independent of the simulation code used.

Fourier space density field can be written [147]:

$$\begin{aligned} \delta(\mathbf{x}) &= \frac{V_\mu}{N}\sum_i^N \delta^D(\mathbf{x}-\mathbf{x}_i) - 1 \Leftrightarrow \\ \delta(\mathbf{k}) &= \frac{1}{V_\mu}\int \mathrm{d}^3\mathbf{x}\delta(\mathbf{x})e^{i\mathbf{k}\cdot\mathbf{x}} \ , \end{aligned} \tag{9.13}$$

where N is the number of particles. The matter power spectra were estimated for each simulation using the standard Fourier based methods [183, 102, 185]: particles and halo centers were interpolated onto a 1024^3 cubical mesh, using the CIC algorithm [92]; the Fast Fourier Transform of the discrete mesh was computed using the FFTW libraries; the power in each Fourier mode was estimated and then corrected for the CIC charge assignment; these estimates were then bin averaged in spherical shells of logarithmic thickness.

Before we proceed to the results, we note that it is not necessarily the case that a simulation that yields a $\lesssim 1\%$ accurate halo mass function should also yield a $\lesssim 1\%$ accurate matter power spectrum, and *vice-versa*. Different requirements for simulation parameters are possible because the mass range in our mass functions only involves a small fraction of the total mass in the simulation. And because, our estimates of the measured power spectra do not extend to scales as small as the virial radius of the smallest halos considered.

Variation with simulation parameters

Figure 9.8 shows the dependence of the matter power spectrum on the simulation run parameters: (top panel) tree-opening angle Θ; (middle panel) the force softening parameter ϵ; (bottom panel) time-step parameter η, for the N-body simulation code `PKDGRAV`. These results show that the estimated power spectra, on large scales ($k/k_{\rm fun} < 10$), are only weakly sensitive to variations in the choice of (Θ, ϵ, η). However, on smaller scales, the power spectra show significant deviations. For the force-softening tests, we find that for $\epsilon \leq l_{\rm m}/5$ the spectra appear to be converged at the sub-percent level on large scales, with a 'bump' at $k/k_{\rm fun} \approx 100$ and a steep drop at smaller scales. This small-scale drop in power is consistent with the puffing up of halo cores that appears to affect the mass function when softening is large. Similarly, for the case of the time-stepping parameter η, we see that for $\eta = 0.6$ there is a $\gtrsim 1\%$ suppression of power for $k/k_{\rm fun} \gtrsim 100$. This can be attributed to the large time-step not being able to follow the rapid changes in the acceleration of particle orbits in the cores of halos – and hence the failure to capture the complex orbit of particles in dense environments. This discussion is limited to `PKDGRAV` run parameters; a detailed study of the dependence of the power spectrum on `Gadget-2` run parameters can be found in Smith et al. [186].

Variation with initial conditions: small boxes

Figure 9.9 shows the variation of the mass power spectra with the choice of 2LPT or 1LPT initial conditions, and with the adopted initial start redshift for the small box simulations

Note that owing to the fact that we are comparing results from the $L = 17.5\,h^{-1}$Mpc at $z = 10$ and $L = 2048\,h^{-1}$Mpc boxes at $z = 0$, we shall refer to wavenumbers in units of the fundamental frequency $k_{\rm fun} = 2\pi/L$.

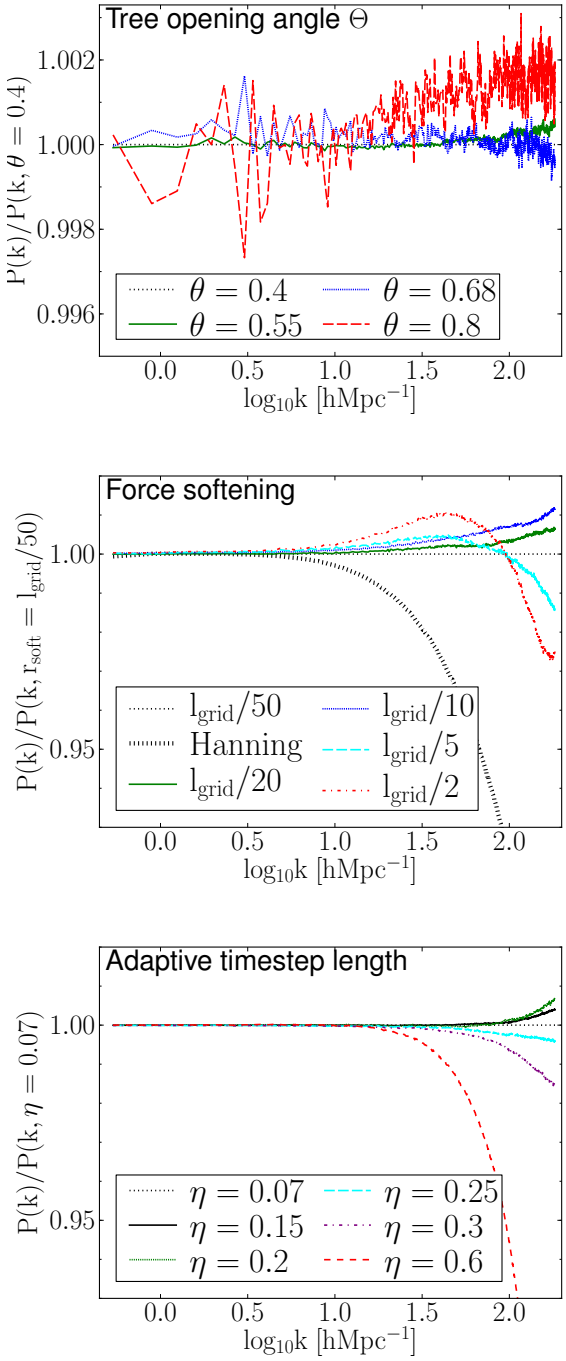


Figure 9.8: Relative variation of the dark matter power spectra with the simulation run parameters for the N-body code PKDGRAV. Top panel: variation with respect to the tree-opening angle Θ. Middle panel: variation with respect to the force softening parameter ϵ. Bottom panel: variation with respect to the time-step parameter η. All panels show the $L = 17.625$ box at $z = 10$. Percent level convergence is seen for each run parameter.

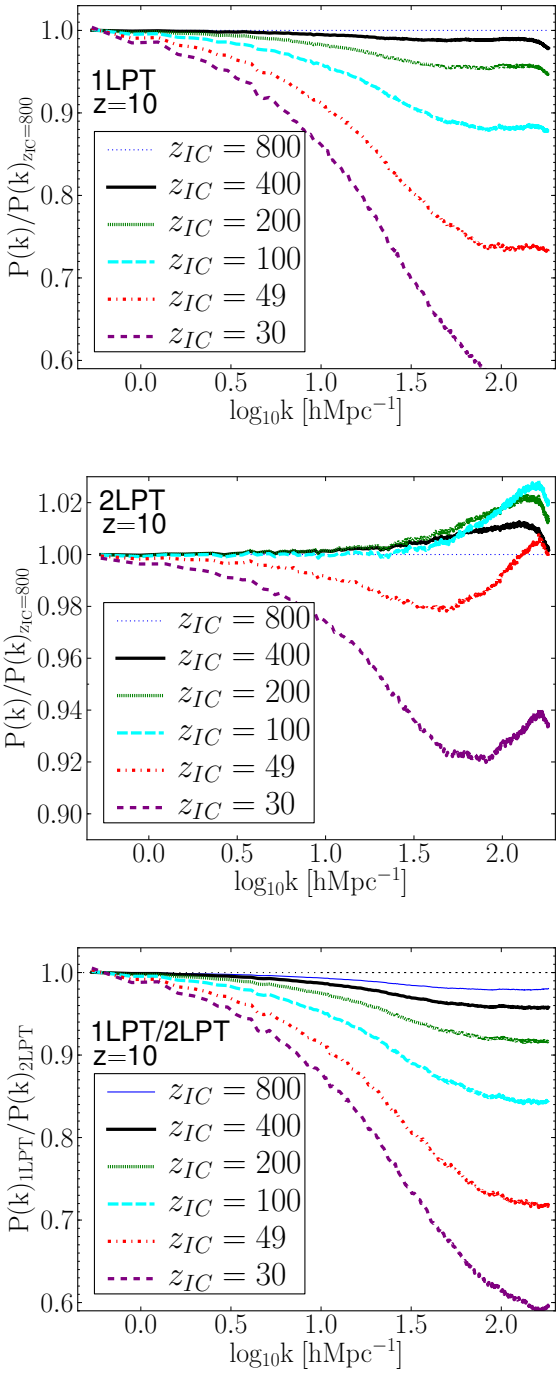


Figure 9.9: Relative variation of the dark matter power spectra with the initial conditions for the N-body code PKDGRAV. Top panel: variation of spectra with initial start redshift for 1LPT initial conditions. Middle panel: same as the top panel except for the case of the 2LPT initial conditions. Bottom panel: ratio of the power spectra from the 1LPT initial condition runs with respect to the 2LPT runs. All results shown are for the $L = 17.625$ box at $z = 10$. Percent level convergence is seen for 2LPT.

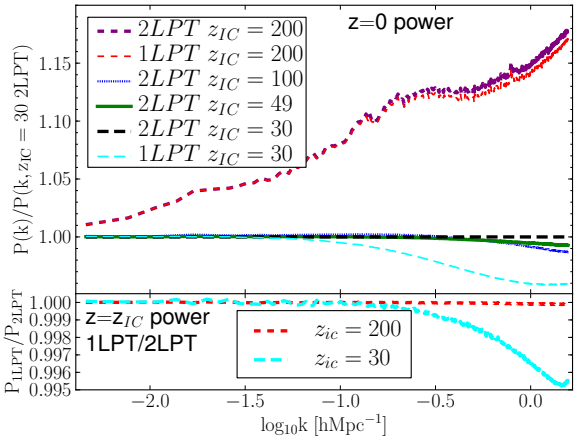


Figure 9.10: Top panel: Ratio of the evolved $z = 0$ matter power spectra for runs with various initial start redshift z_i. The evolved 1LPT and 2LPT power spectra from the early $z_i = 200$ start are very similar to each other, but they lie well above the converged results of lower redshift 2LPT starts. Bottom panel: Initial ratio of the 1LPT to 2LPT matter power spectra at two selected initial start redshifts. Differences in 1LPT and 2LPT power are very small ($< 1\%$) at z_i, then grow much larger as the simulation evolves. The simulations of this figure are large box $L = 2048\,h^{-1}$Mpc simulations run with `PKDGRAV`.

at $z = 10$. The top panel of Fig. 9.9 shows the results for the 1LPT initial conditions. We observe that the power spectra are only converged on the largest scales. On smaller scales, we find that the power increases with increasing start redshift and that the results are almost converged at the percent level only after $a_{\rm f}/a_{\rm i} \gtrsim 80$ expansions.

The middle panel of Fig. 9.9 shows the results for the 2LPT initial conditions. Here we find that simulations that were started with $z_i \gtrsim 100$ are converged for $k/k_{\rm fun} \lesssim 90$. For simulations that possess lower start redshifts we find again a suppression of power, although the effect is much reduced when compared to the equivalent 1LPT start redshift. On smaller scales, $k/k_{\rm fun} \gtrsim 100$, we find that the power is 1% converged for $100 \leq z_i \leq 400$ ($10 \lesssim a_{\rm f}/a_{\rm i} \lesssim 40$), while the $z = 800$ start has up to 2% less power.

The bottom panel of Fig. 9.9 presents the ratio of the power spectra obtained from the 1LPT simulations with the 2LPT power spectra, for various start redshifts. We see that the convergence of the results from the 1LPT simulations with the 2LPT simulations is very slow, and that percent level convergence is only obtained for $z_{\rm i} \gtrsim 800$.

Before continuing, we note that, whilst it appears that percent level convergence in the power spectra may be achieved between 1LPT and 2LPT for very high start redshifts, we have already shown in §9.5 that such high start redshifts are too early to produce an accurate mass function, owing to numerical noise. We are therefore cautious about such convergence.

Variation with initial conditions: large boxes

We now repeat the same set of tests as done for the previous sub-section, only this time we now consider the $L = 2048\,h^{-1}$Mpc simulation cubes at $z = 0$.

Figure 9.10, bottom panel, shows that the 1LPT and 2LPT initial matter power spectra, measured at $z = z_{\rm i}$, are converged at the 1% level with respect to each other for the same start redshift. However, as indicated in the top panel of Fig. 9.10, the evolved spectra started with 1LPT are not converged. On the other hand, the simulations started with the 2LPT initial conditions, appear to be converged at the 1% level for $a_{\rm f}/a_{\rm i} \leq 100$, except perhaps at the smallest scales, even for start redshifts as low as $z_{\rm i} \sim 30$. We note that the evolved 1LPT versus 2LPT simulations, started with $z_{\rm i} = 200$, are significantly discrepant with respect to the other results. As for the case of the mass function, we conjecture that this start redshift is too early for the code `PKDGRAV` to produce an accurate integration of the equations of motion. This reinforces our earlier findings that 1LPT initial conditions are inadequate for accurate simulations.

Several earlier studies have investigated the importance of 1LPT/2LPT initial conditions on the matter power spectrum [39, 87]. Our findings are broadly consistent with these studies. However, Heitmann et al. [87] advocated that 1LPT initial conditions started from $z_{\rm i} = 200$ would lead to better than 1% precision matter power spectra. Clearly such a statement is code and run parameter dependent, and one should be careful of increased numerical errors that may allow consistency between 1LPT and 2LPT while still resulting in inaccurate power spectra.

9.6.2 1-point PDF of matter fluctuations

We now investigate the impact of simulation parameters and 1LPT versus 2LPT initial conditions on the 1-point PDF. At high densities, the 1-point PDF is a useful probe of the central regions of dark matter halos, reflecting many properties of a "stacked" halo density profile [e.g. 167]. There are some technical subtleties as to how one computes the 1-point PDF, since it requires an estimator for the matter density at a given point. The procedure of estimation in general requires one to smooth the particle distribution and hence the results depend up on adopted smoothing scale [see for example 215]. Here we have chosen to compute the 1-point PDF with a 64 particle nearest neighbor kernel. This operates in a similar way to the SPH-kernel and constitutes an adaptive smoothing scale.

Figure 9.11, top panel, shows the variation of the 1-point PDF with the tree-opening angle parameter Θ. We note that the most significant changes are in the regions of highest density, though sensitivity to Θ, beyond the statistical fluctuations, is relatively low. Figure 9.11, middle panel, shows the variation of the 1-point PDF with the force softening ϵ. This clearly shows that the effect of too large force softening is to damp the density distribution in the highest density inner regions of dark matter halos. This "puffs up" halos, which may explain the increased abundances of lower mass halos with increased softening length. We also note that as $\epsilon = l/50$ the dense regions appear to be again suppressed. This we attribute to violent two-body encounters that can evaporate halo cores. Figure 9.11, bottom panel, shows the variation of the 1-point PDF with the time-stepping parameter η. We see that the results are well converged provided $\eta \lesssim 0.15$. The suppression of the high density PDF for large η reinforces our earlier speculation, that if η is too large, then the particle orbits in the cores of halos can not be integrated sufficiently accurately, damping the densities in the inner regions of dark matter halos.

Figure 9.12 presents the 1-point PDF for 1LPT and 2LPT initial conditions for various start redshifts. As can be clearly seen, the results for the 1LPT initial conditions converge

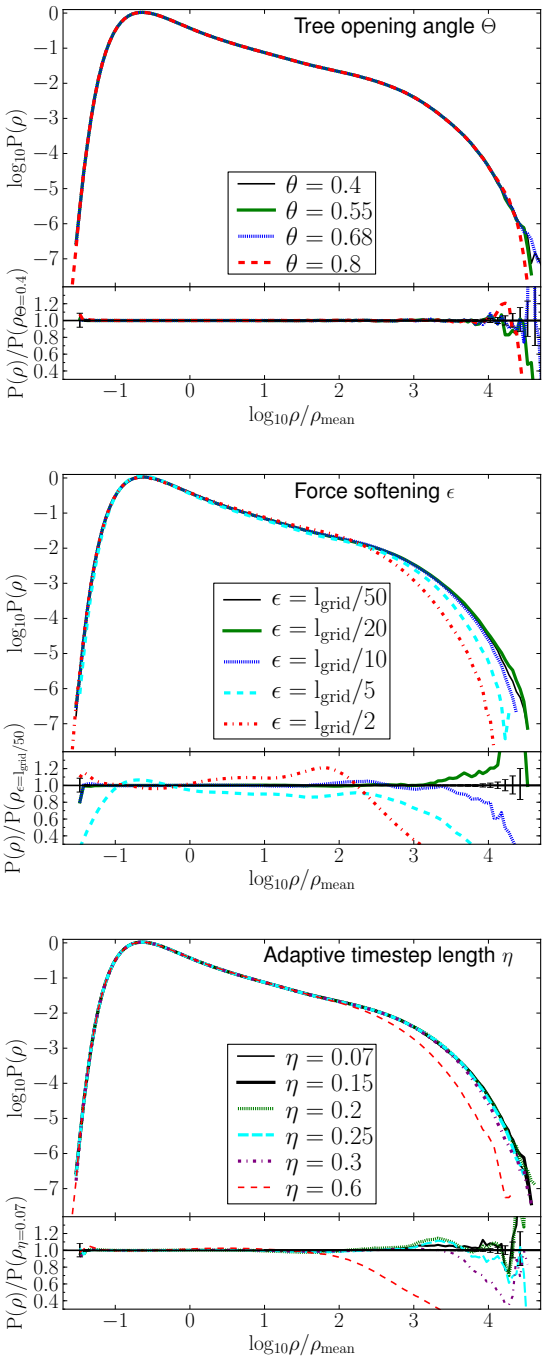


Figure 9.11: Variation of the 1-point PDF of dark matter density fluctuations with the simulation parameters: Top panel, tree-opening angle Θ; Middle panel, force softening ϵ; and Bottom panel, time-stepping parameter η. All results here were obtained from the small $L = 17.625\,h^{-1}$Mpc simulations at $z = 10$ using the tree-code `PKDGRAV`. Sensitivity to numerical parameters is greatest at the highest densities.

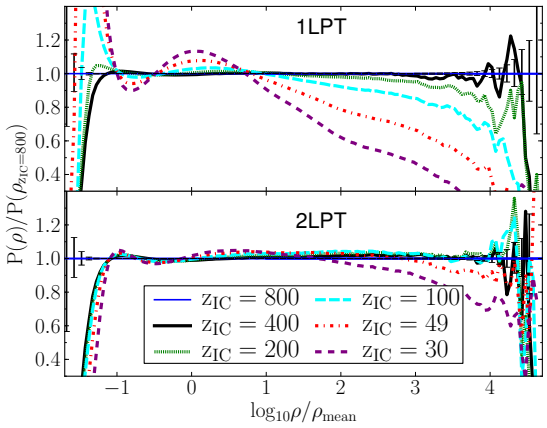


Figure 9.12: Variation of the 1-point PDF of dark matter density fluctuations with respect to the choice of either 1LPT or 2LPT initial conditions for various start redshifts. Top panel: results for the simulations started with 1LPT initial conditions. Bottom panel: shows the same but for the case of 2LPT initial conditions. These results were obtained from the small $L = 17.625\,h^{-1}$Mpc simulations at $z = 10$ using the tree-code `PKDGRAV`.

very slowly with start redshift. We also note that the both the high- and low-density regions appear to be less dense for the simulations that were started with low z_i. For the 2LPT simulations, convergence is reached at much lower start redshifts, roughly $a_f/a_i = 10$ expansion factors of the cube (i.e. around $z_i \sim 100$).

We note all of the converged parameter values, are broadly consistent with those that we identified for the mass function in §9.4, though variations in the PDF at the highest densities are generally larger than 1%.

9.7 Discussion: remaining challenges for <1% accurate mass functions

In this section we discuss the remaining challenges that we will have to face in order to approach better than 1% accurate dark matter halo mass functions.

9.7.1 Mass resolution

In the suite of tests above, we have seen indirect evidence that halos with fewer than $N \sim 1000$ particles, are unlikely to be useful for deriving high accuracy estimates of the mass function. This suggests that there is a critical mass resolution, below which systematic numerical errors are difficult to control. Interestingly, this resolution is somewhat worse than the $N \sim 300$ particle resolution limit expected for a pure particle-mesh (PM) code with mesh spacing equal to the initial inter-particle separation [120]. It is however still better than the more conservative value of $N \sim 2000$ particles proposed by Bhattacharya et al. [26]. Additional support for our claim, comes from the work of Trenti et al. [205], who show through mass-resolution tests, that on a halo-by-halo basis, halo masses with $N \lesssim 1000$ particles are systematically too low. Our statistical limitations mean we can not

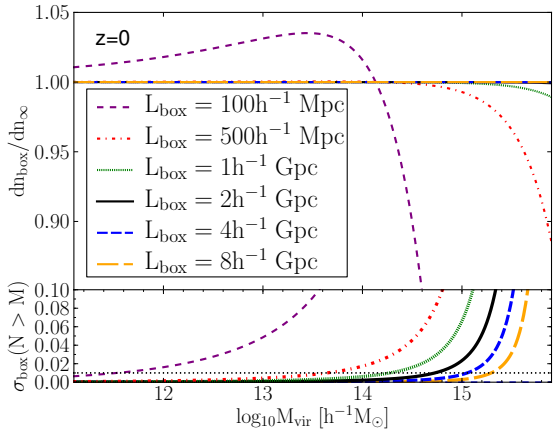


Figure 9.13: Estimates of the relative dependence of the mass function on simulation cube length compared to the case of an infinite cube. The dependence of the mass function on L is calculated by assuming that the power on scales $k < k_{\rm fun} = 2\pi/L$ is zero. The bottom panel is the expected Poisson error of the cumulative halo number count within a single simulation volume. Predictions were obtained using the mass function of Bhattacharya et al. [26].

rule out the possibility that halos resolved with more $N \gtrsim 3000$ particles might be accurate over a larger range in $a_{\rm f}/a_{\rm i}$ than what we find for smaller halos. In a subsequent paper, we will examine directly the mass resolution convergence.

If the critical resolution limit of $N \sim 1000$ particles for accurate halo statistics is upheld, then this suggests that the tree code technique may not have much advantage over the PM code technique in recovering an accurate gravity-only mass function. Of course the higher force resolution for tree codes enables better modelling of higher density regions.

9.7.2 Statistical precision

The full-sky volume out to $z = 2$ is $V_{\mu} \simeq 200\,h^{-3}\,{\rm Gpc}^3$. This sets the requirement on the minimum simulation volume needed to replicate the survey volume accessible to future cluster surveys. This would correspond to a single simulation cube with a side length of roughly $L \gtrsim 3.67\,h^{-1}$Gpc, or if one wants to replicate the full sky-light cone, then one would require a cube of length $L \sim 8\,h^{-1}$Gpc. Obviously performing such a huge simulation with sufficient mass resolution to obtain $N \sim 1000$ particles per halo, for all halos with masses $M \gtrsim 10^{13.5}h^{-1}M_{\odot}$ would be very challenging prospect. Such total volumes may be cheaply covered by combining the results from many smaller volume simulations. However, individual simulation boxes must be large enough to avoid systematic errors due to mode-discreteness near the box-scale and due to the lack of super-box scale power [a "DC-mode" can help for multiple realization ensembles 181].

An estimate of the minimum box size required to avoid suppressing massive halo formation can be made by computing the effect of missing power at wavelengths larger than the box on the (empirically fit) analytic form of the mass function. Figure 9.13 (top panel) demonstrates that a simulation box of roughly $L \sim 2\,h^{-1}$Gpc box should be able to capture the mass function at sub-percent level for all halos with masses $M \lesssim 10^{16}h^{-1}M_{\odot}$ at

$z = 0$. However, in a single realization of such a volume, statistical accuracy will be much lower; Poisson errors remain larger than 1% well below $10^{15}h^{-1}M_{\odot}$. One caveat is that this calculation underestimates the finite volume effects due to the discreteness of modes near the box scale [see for example 162, 184].

9.7.3 Verification of absolute accuracy

The task of verifying that we have actually obtained the true answer is somewhat circular, since it implies that we know already what the true answer is. On our path toward the true answer, we should consider the possibility that our simulations suffer from some level of false convergence. One obvious approach to addressing this issue will be to verify that independent simulation codes give the same results at the desired level of accuracy. However, this does not take into account the pernicious systematic errors, such as the false convergence with redshift that we observed for the 1LPT simulations. In the future, a more complete approach would provide us with a theoretical framework for objectively quantifying 'accuracy' and enable us to identify directions in parameter space that would allow us to approach our desired goal.

There are a number of systematic errors that one will need to characterise in detail. One in particular is that associated with the coarse graining of phase space. Spurious perturbations related to mass discreteness may lead to the collapse of small structures around lattice points [132, 105, 104]. This effect is well-known for Warm Dark Matter simulations [208, 170], and it therefore must also be present in CDM simulations. Other effects relating to mass discreteness in particle codes have been discussed by a number of authors [e.g. 191, 103, 163].

We have attempted to steer away from making direct statements regarding whether particular papers had errors and by how much because quantifying the accuracy of the works of other authors would require repeating their simulations with exactly the same codes and run parameters. A number of widely used fits in the literature used 1LPT initial conditions, many of them with a lower start redshift than expected to be required for good agreement between 1LPT and 2LPT ICs. In those cases, a systematic under-abundance of the halo mass function, especially at high masses or high redshifts is implied.

An informative comparison can be made from the fact that the fitting formulae of multiple authors are in reasonable agreement with each other. For example, there is agreement at the $2-3\%$ level between the FoF mass function fits of Crocce et al. [41, using 2LPT ICs], Bhattacharya et al. [26, using a mix of 2LPT and 1LPT with a high start redshift] and Angulo et al. [8, using 2LPT], over a wide mass range. Also, there is $\simeq 5\%$ agreement between the Tinker et al. [203, 1LPT ICs] and the recent Watson et al. [214, 1LPT ICs with a higher start redshift] spherical over-density mass function fits. There is a caveat that such comparisons are only useful to the extent that the different studies do not suffer from the same systematic errors.

If we consider the widely-used Tinker et al. [203] mass function, the authors stated statistical accuracy of 5% is comparable to the systematic error, estimate from Eqn. 9.11, that we would expect from their results due to their use of 1LPT initial conditions with $a_{\rm f}/a_{\rm i} \sim 50$, although this error would approach 10% at the highest masses, while being smaller at lower masses ($\lesssim 10^{15}h^{-1}M_{\odot}$). They point out some dependence on start redshift

between some of their simulations, and exclude from their fit those with the lowest start redshifts due to a systematic under-abundance of halos. We can thereby deduce that the systematic errors in that study due to initial conditions are generally within their quoted statistical accuracy.

9.7.4 Narrow scale factor range for accuracy

In §9.5, we showed that significant errors are introduced in the halo mass function when the total expansion of the box lies outside the limits ($10 \lesssim a_{\rm f}/a_{\rm i} \lesssim 50$). Of course, one would expect that the epoch of a particular halo's formation is more important than final simulation output for assessing whether that halo, and by extension, the mass function, has been modelled accurately. A brief supporting argument is that any errors introduced to early structures are unlikely to "evolve away", though suppression of the number of early forming halos may become less noticeable, at fixed mass, after more halos have formed at lower redshifts. The implication is that halos in any particular cosmological simulation can be modeled accurately only for a range in formation redshifts of $\Delta_{1+z} \sim 5$. It is possible that this restriction may become less severe as the mass resolution of the simulation is increased. The logic being that more cosmological power at small scales may enable earlier start redshifts to be simulated without leading to an increase in the amount of spurious structure. And, as shown in § 9.5, the mass function sensitivity to start redshift is smaller for halos resolved with more particles. The upper limit of the allowed start redshift range may be code dependent. For example, as we discussed in §9.5.2, a particle-mesh technique may allow higher start redshifts, but typically comes at the cost of worse force resolution.

The narrow range in implied $a_{\rm f}/a_{\rm i}$ presents a challenge for simulations with very large dynamic range, wherein it would be difficult to model accurately the mass function of massive cluster halos forming at $z \sim 0$, while also capturing accurate evolution of the early generations of dwarf galaxy halos forming already at $z \sim 10$. Even though the fraction of mass assembled into galactic halos at such early times is small, early galaxy formation occurs preferentially in Lagrangian regions where clusters will later form. These early-forming galactic halos should be modeled accurately because their feedback processes could have have significant effects on the eventual baryon and total cluster mass content through energy injection and preheating, which could begin very early [e.g. 19].

9.7.5 Impact of baryons

We have purposely ignored the important effects of baryons on the halo mass function. Recent hydrodynamic simulations have shown the range of plausible baryon effects on total cluster halo masses to be up to $\simeq 15\%$ within the radius enclosing $500\times$ the critical density of the universe [197]. Even for "adiabatic simulations" wherein gas cooling, star formation and feedback are ignored, baryons may have up to $\sim 7\%$ effects on the halo mass function [43].

Baryon influences thus present a serious challenge for percent level accuracy in the mass function required for planned dark energy missions. However, accurate gravity-only simulations, as we have explored in this work, are a pre-requisite for future simulations with more complete baryon physics aiming to obtain percent level accuracy in the mass

	Θ	ϵ	η	$\eta_{\rm G}$	$a_{\rm f}/a_{\rm i}$	N per halo	$L\,[h^{-1}{\rm Gpc}]$	$V_\mu(z<2)\,[h^{-3}\,{\rm Gpc}^3]$
min. value	–	$l_{\rm m}/50$	–	–	10	1000	2.0	$\sim$200
max. value	0.7	$l_{\rm m}/20$	0.15-0.20	0.02	40-50	–	–	–

Table 9.2: *Approximate* run and initial condition parameters that permit percent-level simulation convergence in the halo mass function extracted from gravity-only simulations. 2LPT initial conditions are required. Run parameters Θ, ϵ, and η are the tree-opening angle, force softening, time-stepping parameter. $\eta_{\rm G}$ denotes the `Gadget-2` time-step parameter `ErrTolIntAccuracy` $= \eta^2/2$. $a_{\rm f}/a_{\rm i}$ is the ratio of initial to final scale factor at which halo properties are to be considered. L is an estimate of the minimum box length needed to avoid systematic errors in the mass function while V_μ is the comoving volume of the universe accessible to future cluster surveys.

function or other properties. To save computational cost, cosmological constraints might rely on a combination of baryon and gravity-only simulations. For example, hydrodynamic simulations of limited cosmological volumes could be used to calibrate the effects of baryons on halos as well as to derive relations between halos and observable properties. Then, large volume gravity-only simulations might be utilized to map the dependence of halo numbers and other properties on cosmology.

9.7.6 Calibrating mass–observable relationship

A further difficulty in using the halo mass function for cosmology is that any practical halo definition one might use, such as through the FoF or SO algorithms, typically have no direct observable counterpart. For example, the emission from X-ray clusters is determined by the (square of) the gas density distribution. One might argue that this leads to more spherical clusters the abundance of which may be better matched to SO halos in simulations, rather than to FoF halos (see discussion in [112]). Though weak lensing may help to calibrate halo masses [125, 15, 123], optical, x-ray, and Sunyaev Zel'dovich cluster masses have large scatter and systematic uncertainties [8].

Mock observations from simulations may thus be a superior means of obtaining a more accurate mass function in the observable plane, especially once baryon properties are better modeled. Ultimately, an observationally useful and accurate (cluster) mass function will involve modeling baryon physics and observable properties (or mock observations) and represents a formidable challenge for cosmology – a challenge that must be solved in order to fully exploit future and even current cluster surveys of the Universe.

9.8 Conclusions and guidelines for accurate simulations

In this paper we have explored the dependence of the mass function of dark matter halos on simulation run parameters and initial conditions. Our aim has been to perform convergence tests that will illuminate the path to obtaining percent level accuracy in this statistic. This will be a requirement for future cluster surveys of the Universe that aim to help constrain the nature of dark energy or dark gravity.

In §9.2 we gave a brief overview of the simulation method, paying special attention to how one sets up initial conditions, either using Zel'Dovich approximation (1LPT) or 2LPT. We described the simulation codes that we have employed `PKDGRAV` and `Gadget-2`, with the former being the main code used throughout this study.

In §9.3 we described the large suite of N-body simulations that we have performed to study these problems. All simulations were run at $N = 1024^3$ and we covered two regimes: high redshift ($z = 10$), small scale ($L = 17.625\,h^{-1}$Mpc) and low redshift ($z = 0$) large scale ($L = 2\,h^{-1}$Gpc).

In §9.4 we explored the dependence of the mass functions on the simulation parameters and found the following: the resultant mass functions were rather insensitive to the choice of the tree-opening angle, provided $\Theta < 0.7$; the results for halos resolved with fewer than $N \sim 1000$ particles were sensitive to the choice of force softening, with larger values tending to increase the abundance of halos in this regime; results were fairly insensitive to the size of the adaptive time-step parameter and that 1% converged results could be achieved for $\eta \lesssim 0.15$. We also demonstrated that the use of anti-aliasing filters, such as the Hann filter, to set up initial conditions, can lead to $\sim 30\%$ suppression in the abundance of halos resolved with $N \lesssim 1000$ particles. We do not advocate the use of the Hann filter, since *there is no aliasing* in the initial conditions to correct.

In §9.5 we performed a detailed study of the impact of the choice of initial conditions on the mass function. We found that the results from simulations that are initialized with 1LPT converge very slowly as the start redshift is increased. The effect of too low a start redshift being the suppression of the formation of high mass halos. Furthermore, for the large box simulations, we also found simulations started at very high redshifts $z_{\rm i} \gtrsim 200$ would fail to correctly follow the build up of structure due to the relative increase in numerical noise. Furthermore, 1LPT initial conditions exhibit "false convergence" with increasing start redshift. Simulations starting from 2LPT initial conditions proved to have very good convergence properties and for simulations that underwent 10-50 expansion factors, yielded percent level convergence in the halo mass function at the 1024^3 resolution of our tests. We made a direct comparison of these results obtained from integration of the initial conditions with the tree-code `PKDGRAV` with results from the Tree-PM code `Gadget-2`, and found almost identical behaviour. However, a detailed comparison of the mass functions from the two codes revealed that `Gadget-2` produced a $\sim 10\%$ increase in the mass function for halos resolved with $N \lesssim 10^2$ particles. These results extend and support the earlier findings of [39].

In §9.6 we explored the convergence properties of two other statistics of the density field, namely the matter power spectrum and the 1-point probability density function (PDF) of matter fluctuations. We found that the simulation parameters that produced $\lesssim 1\%$

convergence in the mass function would also lead to good convergence behaviour in these statistics. In addition, too high a start redshift for either 1LPT or 2LPT initial conditions would lead to systematic errors. On the other hand, the results from the simulations run with 2LPT initial conditions demonstrated excellent convergence behaviour.

In summary, Table 9.2 presents a general recipe for the parameters needed for percent accuracy of the mass function within a gravity-only simulation using a tree code. Except for the tree opening-angle Θ, which has some dependence on the specific tree used, all the other run parameters can be applicable to other tree codes. This list shows required values but is not complete. In future work, one would expect this table to be extended to include the following: if PM forces are used for large scale force computation, then parameters controlling their accuracy, such as the size of the PM grid should be included; multipole expansions are used to compute the tree forces, and different codes use different orders: which order is sufficiently accurate for our purposes? Also, there should be some entry associated with the parameters that control the halo finder (halo definition).

Ultimately, inferring cosmological parameters from the cluster mass function will require a number of other issues to be solved relating to baryons and observable properties. Among the difficulties that baryons pose is the gravitational coupling of baryon processes to dark matter [187, 46]. Inferring observable properties from the simulations for comparison via mass-observable relations or by direct mock catalogs is a further complexity. Thus, percent level accuracy in numerical simulations represents a formidable challenge, but one that we must meet if future surveys of the Universe are to live up to their potential.

9.9 acknowledgments

We thank the anonymous referee for helpful suggestions that have improved this work. We thank Martin Crocce, Cameron McBride, Roman Scoccimarro, and Romain Teyssier for helpful discussions. We also thank Salman Habib, Katrin Heitmann, and Zarijah Lukić for early discussions that influenced this work. We thank Roman Scoccimarro for making `2LPT` public; Volker Springel for making `GADGET-2` public, and for providing his `B-FoF` halo finder; and Ed Bertschinger for making `GRAFIC-2` public. The simulations were performed on Rosa at the Swiss National Supercomputing Center (CSCS), and the zbox3 and Schrödinger supercomputers at the University of Zurich. RES acknowledges support from a Marie Curie Reintegration Grant and the Alexander von Humboldt Foundation.

Acknowledgements

I have been fascinated with computers from a very early age and was fortunate to have the full support of my parents. When I was twelve years old they bought me a Commodore VIC-20. I quickly moved on from games to actual programming and growing up in a small-ish town in Manitoba it was sometimes difficult to get access to more advanced materials. Whenever the change presented itself to add a book or some software, my parents were always quick to provide it. I suppose that they considered it educational material (which is was), but to me it was a hobby. I will always be grateful for their support.

I would also like to thank David Barnes for providing me with my first job in the field so to speak as an assistant teacher in "Computers for Kids", and for turning over the reins when he moved on. He also introduced me to the science fair which had a positive impact on my education.

By far the greatest influence has been my friend Joachim Stadel. We first met in University and have had a great number of adventures together over the years. Despite being separated, we have always managed to stay in touch, and when he suggested I come to Zürich for even more adventures I leapt at the chance. The work we have done together has been exciting, challenging and very satisfying, and hopefully other have found it useful. I'm sure there is even more in store before we are finally forced to retire and work on building our respective wine cellars.

Ben Moore has also been an inspiration and has always supported me in my endeavours. Romain Teyssier for the interesting work we have done together.

Finally, I would also like to thank Suzanne, Regina, Jonathan, Rebekka, Lea and Simone for their friendship and support over the years.

Bibliography

[1] P. A. R. Ade et al. 'Planck 2013 results. XVI. Cosmological parameters'. In: *Astronomy & Astrophysics* 571 (Nov. 2014), A16. ISSN: 0004-6361. DOI: 10.1051/0004-6361/201321591. arXiv: 1303.5076 (cit. on pp. 71, 103).

[2] P. A. R. Ade et al. 'Planck early results. XI. Calibration of the local galaxy cluster Sunyaev-Zeldovich scaling relations'. In: *Astronomy & Astrophysics* 536 (Dec. 2011), A11. ISSN: 0004-6361. DOI: 10.1051/0004-6361/201116458. arXiv: 1101.2026 (cit. on p. 122).

[3] Oscar Agertz, Romain Teyssier and Ben Moore. 'The formation of disc galaxies in a LCDM universe'. In: *Monthly Notices of the Royal Astronomical Society* 410.2 (Mar. 2010), pp. 1391–1408. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2010.17530.x. arXiv: 1004.0005 (cit. on p. 104).

[4] Jean-Michel Alimi et al. 'DEUS Full Observable LCDM Universe Simulation: the numerical challenge'. In: (June 2012), p. 12. arXiv: 1206.2838 (cit. on pp. 74, 123).

[5] Steven W. Allen, August E. Evrard and Adam B. Mantz. 'Cosmological Parameters from Observations of Galaxy Clusters'. In: *Annual Review of Astronomy and Astrophysics* 49.1 (Sept. 2011), pp. 409–470. ISSN: 0066-4146. DOI: 10.1146/annurev-astro-081710-102514. arXiv: 1103.4829 (cit. on p. 122).

[6] Brandon Allgood et al. 'The shape of dark matter haloes: dependence on mass, redshift, radius and formation'. In: *Monthly Notices of the Royal Astronomical Society* 367.4 (Apr. 2006), pp. 1781–1796. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2006.10094.x. arXiv: 0508497 [astro-ph] (cit. on p. 95).

[7] R. E. Angulo, O Hahn and T Abel. 'How closely do baryons follow dark matter on large scales?' In: *Monthly Notices of the Royal Astronomical Society* 434.2 (Sept. 2013), pp. 1756–1764. ISSN: 0035-8711. DOI: 10.1093/mnras/stt1135. arXiv: 1301.7426 (cit. on p. 115).

[8] R. E. Angulo et al. 'Scaling relations for galaxy clusters in the Millennium-XXL simulation'. In: *Monthly Notices of the Royal Astronomical Society* 426.3 (Mar. 2012), pp. 2046–2062. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2012.21830.x. arXiv: 1203.3216 (cit. on pp. 73, 122, 123, 148, 150).

[9] Itai Arad, Avishai Dekel and Anatoly Klypin. 'Phase-Space Structure of Dark-Matter Haloes: Scale-Invariant PDF Driven by Substructure'. In: *Monthly Notices of the Royal Astronomical Society* 353.1 (Mar. 2004), pp. 15–29. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2004.08045.x. arXiv: 0403135 [astro-ph] (cit. on p. 97).

[10] Yago Ascasibar and J Binney. 'Numerical estimation of densities'. In: *Monthly Notices of the Royal Astronomical Society* 356.3 (Jan. 2005), pp. 872–882. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2004.08480.x. arXiv: 0409233 [astro-ph] (cit. on p. 97).

[11] Vladimir Avila-Reese et al. 'Density profiles of dark matter haloes: diversity and dependence on environment'. In: *Monthly Notices of the Royal Astronomical Society* 310.2 (June 1999), pp. 527–539. ISSN: 00358711. DOI: 10.1046/j.1365-8711.1999.02968.x. arXiv: 9906260 [astro-ph] (cit. on p. 92).

[12] J. S. Bagla and S. Ray. 'Comments on the size of the simulation box in cosmological N-body simulations'. In: *Monthly Notices of the Royal Astronomical Society* 358.3 (Apr. 2005), pp. 1076–1082. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2005.08858.x. arXiv: 0410373 [astro-ph] (cit. on p. 123).

[13] Rennan Barkana and Abraham Loeb. 'Unusually Large Fluctuations in the Statistics of Galaxy Formation at High Redshift'. In: *The Astrophysical Journal* 609.2 (Oct. 2003), pp. 474–481. ISSN: 0004-637X. DOI: 10.1086/421079. arXiv: 0310338 [astro-ph] (cit. on p. 123).

[14] Josh Barnes and Piet Hut. 'A hierarchical O(N log N) force-calculation algorithm'. In: *Nature* 324.6096 (Dec. 1986), pp. 446–449. ISSN: 0028-0836. DOI: 10.1038/324446a0 (cit. on pp. 14, 38, 42, 73).

[15] Matthew R. Becker and Andrey V. Kravtsov. 'On the Accuracy of Weak Lensing Cluster Mass Reconstructions'. In: *The Astrophysical Journal* 740.1 (Nov. 2010), p. 25. ISSN: 0004-637X. DOI: 10.1088/0004-637X/740/1/25. arXiv: 1011.1681 (cit. on p. 150).

[16] Jeroen Bédorf, Evghenii Gaburov and Simon Portegies Zwart. 'Bonsai: A GPU Tree-Code'. In: *ArXiv Astrophysics e-prints* (Apr. 2012), p. 5. arXiv: 1204.2280 (cit. on p. 78).

[17] Jeroen Bédorf et al. '24.77 Pflops on a Gravitational Tree-Code to Simulate the Milky Way Galaxy with 18600 GPUs'. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, p. 54-65* (Dec. 2014), pp. 54–65. DOI: 10.1109/SC.2014.10. arXiv: 1412.0659 (cit. on pp. 74, 75).

[18] Peter S. Behroozi, Risa H. Wechsler and Hao-Yi Wu. 'The Rockstar Phase-Space Temporal Halo Finder and the Velocity Offsets of Cluster Cores'. In: *The Astrophysical Journal* 762.2 (Oct. 2011), p. 109. ISSN: 0004-637X. DOI: 10.1088/0004-637X/762/2/109. arXiv: 1110.4372 (cit. on p. 128).

[19] A. J. Benson and Piero Madau. 'Early Preheating and Galaxy Formation'. In: *Monthly Notices of the Royal Astronomical Society* 344.3 (Mar. 2003), pp. 835–846. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2003.06879.x. arXiv: 0303121 [astro-ph] (cit. on p. 149).

[20] Francis Bernardeau, Martín Crocce and Román Scoccimarro. 'Constructing regularized cosmic propagators'. In: *Physical Review D* 85.12 (June 2012), p. 123519. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.85.123519. arXiv: 1112.3895 [astro-ph.CO] (cit. on p. 118).

[21] F Bernardeau et al. 'Large-scale structure of the Universe and cosmological perturbation theory'. In: *physrep* 367 (Sept. 2002), pp. 1–248. DOI: 10.1016/S0370-1573(02)00135-7 (cit. on p. 105).

[22] Edmund Bertschinger. 'COSMICS: Cosmological Initial Conditions and Microwave Anisotropy Codes'. In: *ArXiv Astrophysics e-prints* (June 1995). arXiv: 9506070 [astro-ph] (cit. on p. 25).

[23] Edmund Bertschinger. 'Multiscale Gaussian Random Fields for Cosmological Simulations'. In: *The Astrophysical Journal Supplement Series* 137.1 (Mar. 2001), p. 38. ISSN: 0067-0049. DOI: 10.1086/322526. arXiv: 0103301 [astro-ph] (cit. on pp. 25, 94, 124, 125).

[24] Edmund Bertschinger and Edmund. 'COSMICS: Cosmological initial conditions and microwave anisotropy codes'. In: *Astrophysics Source Code Library, record ascl:9910.004* (1999) (cit. on p. 124).

[25] Edmund Bertschinger and James M. Gelb. 'Cosmological N-body simulations'. In: *Computers in Physics (ISSN 0894-1866)* 5 (1991), pp. 164–175 (cit. on p. 58).

[26] Suman Bhattacharya et al. 'Mass Function Predictions Beyond LCDM'. In: *The Astrophysical Journal* 732.2 (May 2010), p. 122. ISSN: 0004-637X. DOI: 10.1088/0004-637X/732/2/122. arXiv: 1005.2239 (cit. on pp. 123, 129, 132, 146–148).

[27] J. R. Bond et al. 'Excursion set mass functions for hierarchical Gaussian fluctuations'. In: *The Astrophysical Journal* 379 (Oct. 1991), p. 440. ISSN: 0004-637X. DOI: 10.1086/170520 (cit. on pp. 122, 132).

[28] F. R. Bouchet et al. 'Perturbative Lagrangian Approach to Gravitational Instability'. In: *Astronomy and Astrophysics* (June 1995). arXiv: 9406013 [astro-ph] (cit. on p. 124).

[29] G E P Box and Mervin E Muller. 'A Note on the Generation of Random Normal Deviates'. In: *The Annals of Mathematical Statistics* 29.2 (June 1958), pp. 610–611. ISSN: 0003-4851. DOI: 10.1214/aoms/1177706645 (cit. on pp. 26, 27).

[30] Achi Brandt and Achi. 'Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems'. In: *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*. Vol. 19. Berlin, Heidelberg: Springer Berlin Heidelberg, 1973, pp. 82–89. DOI: 10.1007/BFb0118663 (cit. on p. 74).

[31] T. Buchert, A. L. Melott and A. G. Weiss. 'Optimized Lagrangian Approximations for Modelling Large–Scale Structure at Non–Linear Stages'. In: (Dec. 1994), p. 5. arXiv: 9412075 [astro-ph] (cit. on p. 124).

[32] James S. Bullock et al. 'Profiles of dark haloes: evolution, scatter, and environment'. In: *Monthly Notices of the Royal Astronomical Society* 321.3 (Aug. 1999), pp. 559–575. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2001.04068.x. arXiv: 9908159 [astro-ph] (cit. on pp. 92, 98).

[33] Carmelita Carbone et al. 'Measuring the neutrino mass from future wide galaxy cluster catalogues'. In: *Journal of Cosmology and Astroparticle Physics* 2012.03 (Dec. 2011), pp. 023–023. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2012/03/023. arXiv: 1112.4810 (cit. on p. 122).

[34] Jordan Carlson, Martin White and Nikhil Padmanabhan. 'A critical look at cosmological perturbation theory techniques'. In: *Physical Review D* 80.4 (May 2009), p. 043531. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.80.043531. arXiv: 0905.0479 (cit. on p. 102).

[35] L.-Y. Chiang and P Coles. 'Phase information and the evolution of cosmological density perturbations'. In: *Monthly Notices of the Royal Astronomical Society* 311.4 (Feb. 2000), pp. 809–824. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2000.03086.x (cit. on p. 105).

[36] Stephane Colombi et al. 'Accurate estimators of power spectra in N-body simulations'. In: *Monthly Notices of the Royal Astronomical Society* 393.2 (Nov. 2008), pp. 511–526. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2008.14176.x. arXiv: 0811.0313 (cit. on p. 105).

[37] H. M. P. Couchman, P. A. Thomas and F. R. Pearce. 'Hydra: An Adaptive–Mesh Implementation of PPPM–SPH'. In: *Astrophysical Journal v.452, p.797* 452 (Sept. 1994), p. 797. ISSN: 0004-637X. DOI: 10.1086/176348. arXiv: 9409058 [astro-ph] (cit. on pp. 14, 73).

[38] J. Courtin et al. 'Imprints of dark energy on cosmic structure formation: II) Non-Universality of the halo mass function'. In: *Monthly Notices of the Royal Astronomical Society* 410.3 (Jan. 2010), no–no. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2010.17573.x. arXiv: 1001.3425 (cit. on p. 122).

[39] M. Crocce, S. Pueblas and R. Scoccimarro. 'Transients from initial conditions in cosmological simulations'. In: *Monthly Notices of the Royal Astronomical Society* 373.1 (Nov. 2006), pp. 369–381. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2006.11040.x. arXiv: 0606505 [astro-ph] (cit. on pp. 25, 103, 109, 110, 124, 144, 151).

[40] Martín Crocce and Román Scoccimarro. 'Memory of initial conditions in gravitational clustering'. In: *Physical Review D* 73.6 (Mar. 2006), p. 063520. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.73.063520 (cit. on p. 118).

[41] Martín Crocce et al. 'Simulating the Universe with MICE: the abundance of massive clusters'. In: *Monthly Notices of the Royal Astronomical Society* 403.3 (Apr. 2010), pp. 1353–1367. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2009.16194.x. arXiv: 0907.0019 (cit. on pp. 123, 148).

[42] Weiguang Cui et al. 'An ideal mass assignment scheme for measuring the Power Spectrum with FFTs'. In: *The Astrophysical Journal* 687.2 (Apr. 2008), p. 17. ISSN: 0004-637X. DOI: 10.1086/592079. arXiv: 0804.0070 (cit. on p. 105).

[43] Weiguang Cui et al. 'The effects of baryons on the halo mass function'. In: *Monthly Notices of the Royal Astronomical Society* 423.3 (July 2012), pp. 2279–2287. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2012.21037.x. arXiv: 1111.3066 (cit. on p. 149).

[44] Carlos Cunha, Dragan Huterer and Olivier Dore. 'Primordial non-Gaussianity from the covariance of galaxy cluster counts'. In: (Mar. 2010), p. 15. arXiv: 1003.2416 (cit. on p. 122).

[45] Marcel P. van Daalen and Joop Schaye. 'The contributions of matter inside and outside of haloes to the matter power spectrum'. In: *Monthly Notices of the Royal Astronomical Society* 452.3 (Sept. 2015), pp. 2247–2257. ISSN: 0035-8711. DOI: 10.1093/mnras/stv1456. arXiv: 1501.05950 (cit. on p. 112).

[46] Marcel P. van Daalen et al. 'The effects of galaxy formation on the matter power spectrum: A challenge for precision cosmology'. In: *Monthly Notices of the Royal Astronomical Society* 415.4 (Apr. 2011), pp. 3649–3665. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2011.18981.x. arXiv: 1104.1174 (cit. on pp. 102, 115, 152).

[47] M. Davis et al. 'The evolution of large-scale structure in a universe dominated by cold dark matter'. In: *The Astrophysical Journal* 292 (May 1985), p. 371. ISSN: 0004-637X. DOI: 10.1086/163168 (cit. on pp. 57, 128).

[48] Walter Dehnen. 'A Hierarchical (N) Force Calculation Algorithm'. In: *Journal of Computational Physics* 179.1 (June 2002), pp. 27–42. ISSN: 00219991. DOI: 10.1006/jcph.2002.7026. arXiv: 0202512 [astro-ph] (cit. on pp. 38, 74, 77, 94).

[49] Walter Dehnen. 'A Very Fast and Momentum-Conserving Tree Code'. In: *The Astrophysical Journal* 536.1 (Mar. 2000), pp. L39–L42. ISSN: 0004637X. DOI: 10.1086/312724. arXiv: 0003209 [astro-ph] (cit. on p. 94).

[50] Walter Dehnen. 'Phase-space mixing and the merging of cusps'. In: *Monthly Notices of the Royal Astronomical Society* 360.3 (Apr. 2005), pp. 892–900. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2005.09099.x. arXiv: 0504246 [astro-ph] (cit. on p. 97).

[51] Walter Dehnen. 'Towards optimal softening in three-dimensional N -body codes - I. Minimizing the force error'. In: *Monthly Notices of the Royal Astronomical Society* 324.2 (June 2001), pp. 273–291. ISSN: 00358711. DOI: 10.1046/j.1365-8711.2001.04237.x. arXiv: 0011568 [astro-ph] (cit. on p. 126).

[52] Walter Dehnen and Dean E. McLaughlin. 'Dynamical insight into dark matter haloes'. In: *Monthly Notices of the Royal Astronomical Society* 363.4 (Oct. 2005), pp. 1057–1068. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2005.09510.x. arXiv: 0506528 [astro-ph] (cit. on p. 97).

[53] Walter Dehnen and Justin Read. 'N-body simulations of gravitational dynamics'. In: *The European Physical Journal Plus, Volume 126, article id.55* 126 (May 2011). DOI: 10.1140/epjp/i2011-11055-3. arXiv: 1105.1082 (cit. on p. 76).

[54] Luc Devroye. 'Sample-based non-uniform random variate generation'. In: *Proceedings of the 18th conference on Winter simulation - WSC '86*. WSC '86. New York, New York, USA: ACM Press, 1986, pp. 260–265. ISBN: 0911801111. DOI: 10.1145/318242.318443 (cit. on p. 27).

[55] Juerg Diemand, Piero Madau and Ben Moore. 'The distribution and kinematics of early high- peaks in present-day haloes: implications for rare objects and old stellar populations'. In: *Monthly Notices of the Royal Astronomical Society* 364.2 (Dec. 2005), pp. 367–383. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2005.09604.x. arXiv: 0506615 [astro-ph] (cit. on p. 132).

[56] Juerg Diemand, Ben Moore and Joachim Stadel. 'Convergence and scatter of cluster density profiles'. In: *Monthly Notices of the Royal Astronomical Society* 353.2 (Feb. 2004), pp. 624–632. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2004.08094.x. arXiv: 0402267 [astro-ph] (cit. on p. 92).

[57] Juerg Diemand, Ben Moore and Joachim Stadel. 'Earth-mass dark-matter haloes as the first structures in the early Universe'. In: *Nature* 433.7024 (Jan. 2005), pp. 389–91. ISSN: 1476-4687. DOI: 10.1038/nature03270. arXiv: 0501589 [astro-ph] (cit. on p. 91).

[58] Jurg Diemand, Michael Kuhlen and Piero Madau. 'Dark Matter Substructure and Gamma-Ray Annihilation in the Milky Way Halo'. In: *The Astrophysical Journal* 657.1 (Mar. 2007), pp. 262–270. ISSN: 0004-637X. DOI: 10.1086/510736 (cit. on p. 104).

[59] Jürg Diemand et al. 'Clumps and streams in the local dark matter distribution'. In: *Nature* 454.7205 (Aug. 2008), pp. 735–738. ISSN: 0028-0836. DOI: 10.1038/nature07153. arXiv: 0805.1244 (cit. on pp. 92, 95).

[60] John Dubinski and R. G. Carlberg. 'The structure of cold dark matter halos'. In: *The Astrophysical Journal* 378 (Sept. 1991), p. 496. ISSN: 0004-637X. DOI: 10.1086/170451 (cit. on p. 92).

[61] G. Efstathiou et al. 'Numerical techniques for large cosmological N-body simulations'. In: *The Astrophysical Journal Supplement Series* 57 (Feb. 1985), p. 241. ISSN: 0067-0049. DOI: 10.1086/191003 (cit. on pp. 102, 124).

[62] J Einasto. 'The Andromeda galaxy M31. I. A preliminary model'. In: *Astrofizika* 5 (1969), pp. 137–159 (cit. on p. 95).

[63] Daniel J. Eisenstein and Wayne Hu. 'Baryonic Features in the Matter Transfer Function'. In: *The Astrophysical Journal* 496.2 (Apr. 1998), pp. 605–614. ISSN: 0004-637X. DOI: 10.1086/305424. arXiv: 9709112 [astro-ph] (cit. on p. 128).

[64] Daniel J. Eisenstein and Piet Hut. 'HOP: A New Group-finding Algorithm for N -Body Simulations'. In: *The Astrophysical Journal* 498.1 (May 1998), pp. 137–142. ISSN: 0004-637X. DOI: 10.1086/305535. arXiv: 9712200 [astro-ph] (cit. on p. 59).

[65] P. Fosalba et al. 'The MICE grand challenge lightcone simulation - I. Dark matter clustering'. In: *Monthly Notices of the Royal Astronomical Society* 448.4 (Mar. 2015), pp. 2987–3000. ISSN: 0035-8711. DOI: 10.1093/mnras/stv138. arXiv: 1312.1707 (cit. on p. 102).

[66] P. Fosalba et al. 'The MICE Grand Challenge Lightcone Simulation III: Galaxy lensing mocks from all-sky lensing maps'. In: *Monthly Notices of the Royal Astronomical Society, Volume 447, Issue 2, p.1319-1332* 447 (Dec. 2013), pp. 1319–1332. ISSN: 0035-8711. DOI: 10.1093/mnras/stu2464. arXiv: 1312.2947 (cit. on p. 81).

[67] Matteo Frigo and Steven G Johnson. '{FFTW}: An adaptive software architecture for the {FFT}'. In: *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing*. Vol. 3. IEEE, 1998, pp. 1381–1384 (cit. on p. 28).

[68] Toshiyuki Fukushige and Junichiro Makino. 'Structure of Dark Matter Halos from Hierarchical Clustering'. In: *The Astrophysical Journal* 557.2 (Aug. 2001), pp. 533–545. ISSN: 0004-637X. DOI: 10.1086/321666. arXiv: 0008104 [astro-ph] (cit. on pp. 92, 131).

[69] Sebastiano Ghigna et al. 'Dark Matter Halos within Clusters'. In: *Monthly Notices of the Royal Astronomical Society* 300.1 (Jan. 1998), pp. 146–162. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.1998.01918.x. arXiv: 9801192 [astro-ph] (cit. on p. 92).

[70] Nickolay Y. Gnedin, Andrey V. Kravtsov and Douglas H. Rudd. 'Implementing the DC Mode in Cosmological Simulations with Supercomoving Variables'. In: *The Astrophysical Journal Supplement Series* 194.2 (Apr. 2011), p. 46. ISSN: 0067-0049. DOI: 10.1088/0067-0049/194/2/46. arXiv: 1104.1428 (cit. on p. 111).

[71] Tobias Goerdt et al. 'Core creation in galaxies and haloes via sinking massive objects'. In: *The Astrophysical Journal* 725.2 (June 2008), pp. 1707–1716. ISSN: 0004-637X. DOI: 10.1088/0004-637X/725/2/1707. arXiv: 0806.1951 (cit. on p. 92).

[72] K. M. Gorski et al. 'HEALPix – a Framework for High Resolution Discretization, and Fast Analysis of Data Distributed on the Sphere'. In: *The Astrophysical Journal* 622.2 (Sept. 2004), pp. 759–771. ISSN: 0004-637X. DOI: 10.1086/427976. arXiv: 0409513 [astro-ph] (cit. on p. 15).

[73] Stefan Gottlöber, Anatoly Klypin and Andrey V. Kravtsov. 'Halo evolution in a cosmological environment'. In: *Observational Cosmology: The Development of Galaxy Systems* 176 (1999) (cit. on p. 58).

[74] Fabio Governato et al. 'The Local Group as a test of cosmological models'. In: *New Astronomy* 2.2 (July 1997), pp. 91–106. ISSN: 13841076. DOI: 10.1016/S1384-1076(97)00011-0. arXiv: 9612007 [astro-ph] (cit. on pp. 58, 59).

[75] L Greengard and V Rokhlin. 'A fast algorithm for particle simulations'. In: *Journal of Computational Physics* 73.2 (Dec. 1987), pp. 325–348. ISSN: 00219991. DOI: 10.1016/0021-9991(87)90140-9 (cit. on pp. 14, 74, 75).

[76] Simon L. Grimm and Joachim G. Stadel. 'The GENGA Code: Gravitational Encounters in N-body simulations with GPU Acceleration'. In: (Apr. 2014). DOI: 10.1088/0004-637X/796/1/23. arXiv: 1404.2324 (cit. on p. 14).

[77] James E. Gunn and III Gott, J. Richard. 'On the Infall of Matter Into Clusters of Galaxies and Some Effects on Their Evolution'. In: *The Astrophysical Journal* 176 (Aug. 1972), p. 1. ISSN: 0004-637X. DOI: 10.1086/151605 (cit. on p. 58).

[78] Salman Habib et al. 'HACC'. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on - SC '13*. New York, New York, USA: ACM Press, 2013, pp. 1–10. ISBN: 9781450323789. DOI: 10.1145/2503210.2504566 (cit. on pp. 74, 84).

[79] Salman Habib et al. 'HACC: Simulating Sky Surveys on State-of-the-Art Supercomputing Architectures'. In: *New Astronomy, Volume 42, p. 49-65.* 42 (Oct. 2014), pp. 49–65. ISSN: 1384-1076. DOI: 10.1016/j.newast.2015.06.003. arXiv: 1410.2805 (cit. on p. 84).

[80] Oliver Hahn and Tom Abel. 'Multi-scale initial conditions for cosmological simulations'. In: *Monthly Notices of the Royal Astronomical Society* 415.3 (Aug. 2011), pp. 2101–2121. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2011.18820.x. arXiv: 1103.6031 (cit. on pp. 25, 109).

[81] Zoltan Haiman, Joseph J. Mohr and Gilbert P. Holder. 'Constraints on Cosmological Parameters from Future Galaxy Cluster Surveys'. In: *The Astrophysical Journal* 553.2 (June 2001), pp. 545–561. ISSN: 0004-637X. DOI: 10.1086/320939. arXiv: 0002336 [astro-ph] (cit. on p. 122).

[82] Steen H. Hansen and Joachim Stadel. 'The velocity anisotropy—density slope relation'. In: *Journal of Cosmology and Astroparticle Physics* 2006.05 (May 2006), pp. 014–014. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2006/05/014. arXiv: 0510656 [astro-ph] (cit. on p. 92).

[83] J. Heinrich. *Coverage of Error Bars for Poisson Data*. Tech. rep. CDF Note 6438 Batavia: Fermilab, 2003 (cit. on p. 129).

[84] Katrin Heitmann et al. 'Robustness of Cosmological Simulations. I. Large-Scale Structure'. In: *The Astrophysical Journal Supplement Series* 160.1 (Sept. 2005), pp. 28–58. ISSN: 0067-0049. DOI: 10.1086/432646. arXiv: 0411795 [astro-ph] (cit. on p. 102).

[85] Katrin Heitmann et al. 'The cosmic code comparison project'. In: *Computational Science & Discovery* 1.1 (Nov. 2008), p. 015003. ISSN: 1749-4699. DOI: 10.1088/1749-4699/1/1/015003. arXiv: 0706.1270 (cit. on pp. 102, 106, 114, 137).

[86] Katrin Heitmann et al. 'The Coyote Universe Extended: Precision Emulation of the Matter Power Spectrum'. In: *The Astrophysical Journal* 780.1 (Apr. 2013), p. 111. ISSN: 0004-637X. DOI: 10.1088/0004-637X/780/1/111. arXiv: 1304.7849 (cit. on pp. 113, 114).

[87] Katrin Heitmann et al. 'The Coyote Universe I: Precision Determination of the Nonlinear Matter Power Spectrum'. In: *The Astrophysical Journal* 715.1 (Dec. 2008), pp. 104–121. ISSN: 0004-637X. DOI: 10.1088/0004-637X/715/1/104. arXiv: 0812.1052 (cit. on pp. 103, 106, 109, 112, 114, 144).

[88] Katrin Heitmann et al. 'The Q Continuum Simulation: Harnessing the Power of GPU Accelerated Supercomputers'. In: *The Astrophysical Journal Supplement Series, Volume 219, Issue 2, article id. 34, 13 pp. (2015).* 219 (Nov. 2014). ISSN: 0067-0049. DOI: 10.1088/0067-0049/219/2/34. arXiv: 1411.3396 (cit. on p. 74).

[89] Lars Hernquist. 'An analytical model for spherical galaxies and bulges'. In: *The Astrophysical Journal* 356 (June 1990), p. 359. ISSN: 0004-637X. DOI: 10.1086/168845 (cit. on p. 95).

[90] Lars Hernquist, Francois R. Bouchet and Yasushi Suto. 'Application of the Ewald method to cosmological N-body simulations'. In: *The Astrophysical Journal Supplement Series* 75 (Feb. 1991), p. 231. ISSN: 0067-0049. DOI: 10.1086/191530 (cit. on pp. 39, 77).

[91] Lars Hernquist and Neal Katz. 'TREESPH - A unification of SPH with the hierarchical tree method'. In: *The Astrophysical Journal Supplement Series* 70 (June 1989), p. 419. ISSN: 0067-0049. DOI: 10.1086/191344 (cit. on p. 59).

[92] R. W. Hockney and J. W. Eastwood. 'Computer simulation using particles'. In: *Bristol: Hilger* (1988) (cit. on pp. 14, 73, 140).

[93] Yehuda Hoffman and Erez Ribak. 'Constrained realizations of Gaussian fields - A simple algorithm'. In: *The Astrophysical Journal* 380 (Oct. 1991), p. L5. ISSN: 0004-637X. DOI: 10.1086/186160 (cit. on p. 26).

[94] Erik Holmberg and Erik. 'On the Clustering Tendencies among the Nebulae. II. a Study of Encounters Between Laboratory Models of Stellar Systems by a New Integration Procedure.' In: *The Astrophysical Journal* 94 (Nov. 1941), p. 385. ISSN: 0004-637X. DOI: 10.1086/144344 (cit. on pp. 14, 73).

[95] Ilian T. Iliev et al. 'Can 21-cm observations discriminate between high-mass and low-mass galaxies as reionization sources?' In: *Monthly Notices of the Royal Astronomical Society* 423.3 (July 2012), pp. 2222–2253. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2012.21032.x. arXiv: 1107.4772 (cit. on p. 123).

[96] Tomoaki Ishiyama, Toshiyuki Fukushige and Junichiro Makino. 'GreeM : Massively Parallel TreePM Code for Large Cosmological N-body Simulations'. In: *Publications of the Astronomical Society of Japan, Vol.61, No.6, pp.1319–1330* 61 (Oct. 2009), pp. 1319–1330. ISSN: 0004-6264. DOI: 10.1093/pasj/61.6.1319. arXiv: 0910.0121 (cit. on p. 74).

[97] Tomoaki Ishiyama, Keigo Nitadori and Junichiro Makino. '4.45 Pflops Astrophysical N-Body Simulation on K computer – The Gravitational Trillion-Body Problem'. In: *eprint arXiv:1211.4406* (Nov. 2012). arXiv: 1211.4406 (cit. on p. 74).

[98] Z. Ivezic et al. 'LSST: from Science Drivers to Reference Design and Anticipated Data Products'. In: (May 2008). arXiv: 0805.2366 (cit. on pp. 102, 112).

[99] Adrian Jenkins. 'Second-order Lagrangian perturbation theory initial conditions for resimulations'. In: *Monthly Notices of the Royal Astronomical Society* 403.4 (Apr. 2010), pp. 1859–1872. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2010.16259.x. arXiv: 0910.0258 (cit. on pp. 109, 125).

[100] A. Jenkins et al. 'Mass function of dark matter halos'. In: *Monthly Notices of the Royal Astronomical Society* 321.2 (May 2000), pp. 372–384. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2001.04029.x. arXiv: 0005260 [astro-ph] (cit. on p. 123).

[101] Y. P. Jing and Yasushi Suto. 'Triaxial Modeling of Halo Density Profiles with High-resolution N-body Simulations'. In: *The Astrophysical Journal* 574.2 (Feb. 2002), pp. 538–553. ISSN: 0004-637X. DOI: 10.1086/341065. arXiv: 0202064 [astro-ph] (cit. on p. 95).

[102] Y.˜P. P. Jing. 'Correcting for the Alias Effect When Measuring the Power Spectrum Using a Fast Fourier Transform'. In: *The Astrophysical Journal* 620.2 (Feb. 2005), pp. 559–563. ISSN: 0004-637X. DOI: 10.1086/427087. arXiv: 0409240 [astro-ph] (cit. on pp. 105, 140).

[103] M. Joyce and B. Marcos. 'Quantification of discreteness effects in cosmological N-body simulations. II. Evolution up to shell crossing'. In: *Physical Review D* 76.10 (Nov. 2007), p. 103505. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.76.103505. arXiv: 0704.3697 (cit. on p. 148).

[104] M. Joyce, B. Marcos and T. Baertschiger. 'Towards quantitative control on discreteness error in the non-linear regime of cosmological N -body simulations'. In: *Monthly Notices of the Royal Astronomical Society* 394.2 (Apr. 2009), pp. 751–773. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2008.14290.x. arXiv: 0805.1357 (cit. on pp. 112, 148).

[105] Michael Joyce and Bruno Marcos. 'Quantification of discreteness effects in cosmological N-body simulations: I. Initial Conditions'. In: *Physical Review D* 75.6 (Oct. 2004), p. 063516. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.75.063516. arXiv: 0410451 [astro-ph] (cit. on p. 148).

[106] Ralf Klessen and Ralf. 'GRAPESPH with Fully Periodic Boundaries: Fragmentation of Molecular Clouds'. In: *Computational Astrophysics; 12th Kingston Meeting on Theoretical Astrophysics; proceedings of meeting held in Halifax; Nova Scotia; Canada October 17-19; 1996, ASP Conference Series #123, edited by D. A. Clarke and M. J. West., p. 169* 123 (1997), p. 169. ISSN: 1050-3390 (cit. on p. 77).

[107] Alexander Knebe et al. 'Haloes gone MAD: The Halo-Finder Comparison Project'. In: *Monthly Notices of the Royal Astronomical Society* 415.3 (Apr. 2011), pp. 2293–2318. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2011.18858.x. arXiv: 1104.0949 (cit. on pp. 57, 128).

[108] Alexander Knebe et al. 'On the starting redshift for cosmological simulations: Focusing on halo properties'. In: *The Astrophysical Journal* 698.1 (Apr. 2009), pp. 266–274. ISSN: 0004-637X. DOI: 10.1088/0004-637X/698/1/266. arXiv: 0904.0083 (cit. on pp. 110, 124).

[109] Alexander Knebe et al. 'Structure Finding in Cosmological Simulations: The State of Affairs'. In: *Monthly Notices of the Royal Astronomical Society* 435.2 (Apr. 2013), pp. 1618–1658. ISSN: 0035-8711. DOI: 10.1093/mnras/stt1403. arXiv: 1304.0585 (cit. on p. 57).

[110] E. Komatsu et al. 'Five-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation'. In: *The Astrophysical Journal Supplement Series* 180.2 (Mar. 2008), pp. 330–376. ISSN: 0067-0049. DOI: 10.1088/0067-0049/180/2/330. arXiv: 0803.0547 (cit. on p. 128).

[111] E. Komatsu et al. 'Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation'. In: *The Astrophysical Journal Supplement Series* 192.2 (Jan. 2010), p. 18. ISSN: 0067-0049. DOI: 10.1088/0067-0049/192/2/18. arXiv: 1001.4538 (cit. on pp. 122, 128).

[112] Andrey Kravtsov and Stefano Borgani. 'Formation of Galaxy Clusters'. In: (May 2012), p. 66. DOI: 10.1146/annurev-astro-081811-125502. arXiv: 1205.5556 (cit. on p. 150).

[113] Cedric Lacey and Shaun Cole. 'Merger rates in hierarchical models of galaxy formation. II: Comparison with N-body simulations'. In: (Feb. 1994). DOI: 10.1093/mnras/271.3.676. arXiv: 9402069 [astro-ph] (cit. on p. 128).

[114] Ofer Lahav and Andrew R Liddle. 'The Cosmological Parameters 2006'. In: *Journal of Physics G: Nuclear and Particle Physics* 33.1 (Jan. 2006), pp. 1–1232. ISSN: 0954-3899. DOI: 10.1088/0954-3899/33/1/001. arXiv: 0601168 [astro-ph] (cit. on p. 92).

[115] R. Laureijs et al. 'Euclid Definition Study Report'. In: *eprint arXiv:1110.3193* (Oct. 2011), p. 116. arXiv: 1110.3193 (cit. on pp. 71, 102, 112, 122).

[116] Antony Lewis, Anthony Challinor and Anthony Lasenby. 'Efficient Computation of CMB anisotropies in closed FRW models'. In: *The Astrophysical Journal* 538.2 (Nov. 1999), pp. 473–476. ISSN: 0004-637X. DOI: 10.1086/309179. arXiv: 9911177 [astro-ph] (cit. on p. 128).

[117] Benjamin L'Huillier, Changbom Park and Juhan Kim. 'Effects of the initial conditions on cosmological N-body simulations'. In: *New Astronomy* 30 (July 2014), pp. 79–88. ISSN: 13841076. DOI: 10.1016/j.newast.2014.01.007. arXiv: 1401.6180 (cit. on p. 110).

[118] Marcos Lima and Wayne Hu. 'Self-Calibration of Cluster Dark Energy Studies: Observable-Mass Distribution'. In: *Physical Review D* 72.4 (Mar. 2005), p. 043006. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.72.043006. arXiv: 0503363 [astro-ph] (cit. on p. 122).

[119] LSST Science Collaboration et al. 'LSST Science Book, Version 2.0'. In: (Dec. 2009). arXiv: 0912.0201 (cit. on p. 71).

[120] Zarija Lukic et al. 'The Halo Mass Function: High-Redshift Evolution and Universality'. In: *The Astrophysical Journal* 671.2 (Feb. 2007), pp. 1160–1181. ISSN: 0004-637X. DOI: 10.1086/523083. arXiv: 0702360 [astro-ph] (cit. on pp. 129, 132, 146).

[121] Zarija Lukić et al. 'The Structure of Halos: Implications for Group and Cluster Cosmology'. In: *The Astrophysical Journal* 692.1 (Mar. 2008), pp. 217–228. ISSN: 0004-637X. DOI: 10.1088/0004-637X/692/1/217. arXiv: 0803.3624 (cit. on p. 128).

[122] Andrea V. Maccio' et al. 'Concentration, Spin and Shape of Dark Matter Haloes: Scatter and the Dependence on Mass and Environment'. In: *Monthly Notices of the Royal Astronomical Society* 378.1 (Aug. 2006), pp. 55–71. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2007.11720.x. arXiv: 0608157 [astro-ph] (cit. on p. 98).

[123] Rachel Mandelbaum et al. 'Precision cluster mass determination from weak lensing'. In: *Monthly Notices of the Royal Astronomical Society* (Nov. 2009), no–no. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2010.16619.x. arXiv: 0911.4972 (cit. on pp. 122, 150).

[124] Laura Marian and Gary M. Bernstein. 'Dark energy constraints from lensing-detected galaxy clusters'. In: *Physical Review D* 73.12 (June 2006), p. 123525. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.73.123525. arXiv: 0605746 [astro-ph] (cit. on p. 122).

[125] Laura Marian, Robert E. Smith and Gary M. Bernstein. 'The cosmology dependence of weak lensing cluster counts'. In: *The Astrophysical Journal* 698.1 (Nov. 2008), pp. L33–L36. ISSN: 0004-637X. DOI: 10.1088/0004-637X/698/1/L33. arXiv: 0811.1991 (cit. on pp. 122, 150).

[126] Laura Marian et al. 'Measuring primordial non-Gaussianity through weak lensing peak counts'. In: *The Astrophysical Journal* 728.1 (Oct. 2010), p. L13. ISSN: 2041-8205. DOI: 10.1088/2041-8205/728/1/L13. arXiv: 1010.5242 (cit. on p. 122).

[127] George Marsaglia and Arif Zaman. 'A New Class of Random Number Generators'. In: *The Annals of Applied Probability* 1.3 (Aug. 1991), pp. 462–480. ISSN: 1050-5164. DOI: 10.1214/aoap/1177005878 (cit. on p. 27).

[128] Sabino Matarrese, Licia Verde and Raul Jimenez. 'The abundance of high-redshift objects as a probe of non-Gaussian initial conditions'. In: *The Astrophysical Journal* 541.1 (Jan. 2000), pp. 10–24. ISSN: 0004-637X. DOI: 10.1086/309412. arXiv: 0001366 [astro-ph] (cit. on p. 122).

[129] Makoto Matsumoto and Takuji Nishimura. 'Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator'. In: *ACM Transactions on Modeling and Computer Simulation* 8.1 (Jan. 1998), pp. 3–30. ISSN: 10493301. DOI: 10.1145/272991.272995 (cit. on p. 27).

[130] Cameron McBride et al. 'LasDamas: Accurate Determination of the Abundance of Galaxy Clusters'. In: *American Astronomical Society* 43 (2011) (cit. on p. 128).

[131] James McBride, Onsi Fakhouri and Chung-Pei Ma. 'Mass Accretion Rates and Histories of Dark Matter Haloes'. In: *Monthly Notices of the Royal Astronomical Society* 398.4 (Feb. 2009), pp. 1858–1868. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2009.15329.x. arXiv: 0902.3659 (cit. on p. 110).

[132] Adrian L. Melott. 'More Resolution Isn't Always Better Resolution'. In: *Communications in Astrophysics* 15 (1990) (cit. on p. 148).

[133] David Merritt et al. 'Empirical models for Dark Matter Halos. I. Nonparametric Construction of Density Profiles and Comparison with Parametric Models'. In: *The Astronomical Journal* 132.6 (Sept. 2005), pp. 2685–2700. ISSN: 0004-6256. DOI: 10.1086/508988. arXiv: 0509417 [astro-ph] (cit. on p. 95).

[134] Michael S. Warren et al. 'Pentium Pro Inside: I. A Treecode at 430 Gigaflops on ASCI Red, II. Price/Performance of $50/Mflop on Loki and Hyglac'. In: *SC Conference* 00.undefined (1997), p. 61. DOI: doi.ieeecomputersociety.org/10.1109/SC.1997.10057 (cit. on p. 73).

[135] Irshad Mohammed and Uros Seljak. 'Analytic model for the matter power spectrum, its covariance matrix, and baryonic effects'. In: *Monthly Notices of the Royal Astronomical Society, Volume 445, Issue 4, p.3382-3400* 445 (June 2014), pp. 3382–3400. ISSN: 0035-8711. DOI: 10.1093/mnras/stu1972. arXiv: 1407.0060 (cit. on pp. 102, 111).

[136] Ben Moore et al. 'Cold collapse and the core catastrophe'. In: *Monthly Notices of the Royal Astronomical Society* 310.4 (Mar. 1999), pp. 1147–1152. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.1999.03039.x. arXiv: 9903164 [astro-ph] (cit. on p. 92).

[137] Ben Moore et al. 'Dark matter in Draco and the Local Group: Implications for direct detection experiments'. In: *Physical Review D* 64.6 (Aug. 2001), p. 063508. ISSN: 0556-2821. DOI: 10.1103/PhysRevD.64.063508 (cit. on p. 92).

[138] Ben Moore et al. 'Resolving the Structure of Cold Dark Matter Halos'. In: *The Astrophysical Journal* 499.1 (May 1998), pp. L5–L8. ISSN: 0004637X. DOI: 10.1086/311333. arXiv: 9709051 [astro-ph] (cit. on pp. 92, 131).

[139] Surhud More et al. 'The overdensity and masses of the friends-of-friends halos and universality of the halo mass function'. In: *The Astrophysical Journal Supplement Series* 195.1 (Feb. 2011), p. 4. ISSN: 0067-0049. DOI: 10.1088/0067-0049/195/1/4. arXiv: 1103.0005 (cit. on p. 128).

[140] Julio F. Navarro, Carlos S. Frenk and Simon D. M. White. 'The Structure of Cold Dark Matter Halos'. In: *The Astrophysical Journal* 462.2 (Aug. 1995), p. 563. ISSN: 0004-637X. DOI: 10.1086/177173. arXiv: 9508025 [astro-ph] (cit. on p. 92).

[141] Julio F. Navarro et al. 'The Inner Structure of LambdaCDM Halos III: Universality and Asymptotic Slopes'. In: *Monthly Notices of the Royal Astronomical Society* 349.3 (Nov. 2003), pp. 1039–1051. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2004.07586.x. arXiv: 0311231 [astro-ph] (cit. on pp. 92, 95).

[142] P Ocvirk, C Pichon and R Teyssier. 'Bimodal gas accretion in the Horizon-MareNostrum galaxy formation simulation'. In: *Monthly Notices of the Royal Astronomical Society* 390 (Mar. 2008), pp. 1326–1338. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2008.13763.x. arXiv: 0803.4506 (cit. on p. 104).

[143] Masamune Oguri and Masahiro Takada. 'Combining cluster observables and stacked weak lensing to probe dark energy: Self-calibration of systematic uncertainties'. In: *Physical Review D* 83.2 (Oct. 2010), p. 023008. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.83.023008. arXiv: 1010.0744 (cit. on p. 122).

[144] Julian Onions et al. 'SubHaloes going Notts: The SubHalo-Finder Comparison Project'. In: *Monthly Notices of the Royal Astronomical Society* 423.2 (Mar. 2012), pp. 1200–1214. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2012.20947.x. arXiv: 1203.3695 (cit. on p. 57).

[145] Brian W. O'Shea et al. 'Comparing AMR and SPH Cosmological Simulations. I. Dark Matter and Adiabatic Simulations'. In: *The Astrophysical Journal Supplement Series* 160.1 (Sept. 2005), pp. 1–27. ISSN: 0067-0049. DOI: 10.1086/432645. arXiv: 0312651 [astro-ph] (cit. on p. 102).

[146] P. J. E. Peebles. 'Large-scale background temperature and mass fluctuations due to scale-invariant primeval perturbations'. In: *The Astrophysical Journal* 263 (Dec. 1982), p. L1. ISSN: 0004-637X. DOI: 10.1086/183911 (cit. on p. 91).

[147] P. J. E. Peebles. 'The large-scale structure of the universe'. In: *Research supported by the National Science Foundation. Princeton* (1980) (cit. on p. 140).

[148] P. J. E. Peebles and P. J. E. 'Structure of the Coma Cluster of Galaxies'. In: *The Astronomical Journal* 75 (Feb. 1970), p. 13. ISSN: 00046256. DOI: 10.1086/110933 (cit. on pp. 14, 73).

[149] Ue-Li Pen. 'Reconstructing Nonlinear Stochastic Bias from Velocity Space Distortions'. In: *The Astrophysical Journal* 504.2 (Sept. 1998), pp. 601–606. ISSN: 0004-637X. DOI: 10.1086/306098 (cit. on p. 105).

[150] Annalisa Pillepich, Cristiano Porciani and Thomas H. Reiprich. 'The X-ray cluster survey with : forecasts for cosmology, cluster physics and primordial non-Gaussianity'. In: *Monthly Notices of the Royal Astronomical Society* 422.1 (May 2012), pp. 44–69. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2012.20443.x. arXiv: 1111.6587 (cit. on p. 122).

[151] Douglas Potter, Joachim Stadel and Romain Teyssier. 'PKDGRAV3: Beyond Trillion Particle Cosmological Simulations for the Next Era of Galaxy Surveys'. In: (Sept. 2016). arXiv: 1609.08621 (cit. on pp. 15, 25).

[152] Chris Power and Alexander Knebe. 'The Impact of Box Size on the Properties of Dark Matter Haloes in Cosmological Simulations'. In: *Monthly Notices of the Royal Astronomical Society* 370.2 (Dec. 2005), pp. 691–701. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2006.10562.x. arXiv: 0512281 [astro-ph] (cit. on p. 123).

[153] C. Power et al. 'The Inner Structure of LambdaCDM Halos I: A Numerical Convergence Study'. In: *Monthly Notices of the Royal Astronomical Society* 338.1 (Jan. 2002), pp. 14–34. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2003.05925.x. arXiv: 0201544 [astro-ph] (cit. on pp. 92, 131, 132).

[154] William H. Press and Paul Schechter. 'Formation of Galaxies and Clusters of Galaxies by Self-Similar Gravitational Condensation'. In: *The Astrophysical Journal* 187 (Feb. 1974), p. 425. ISSN: 0004-637X. DOI: 10.1086/152650 (cit. on p. 122).

[155] William H. Press et al. *Numerical recipes in FORTRAN. The art of scientific computing*. 1992 (cit. on p. 125).

[156] P Prugniel and F Simien. 'The fundamental plane of early-type galaxies: non-homology of the spatial structure.' In: *aap* 321 (May 1997), pp. 111–122 (cit. on p. 95).

[157] S. Prunet et al. 'Initial Conditions for Large Cosmological Simulations'. In: *The Astrophysical Journal Supplement Series* 178.2 (Apr. 2008), pp. 179–188. ISSN: 0067-0049. DOI: 10.1086/590370. arXiv: 0804.3536 (cit. on p. 25).

[158] Yann Rasera et al. 'Cosmic-variance limited Baryon Acoustic Oscillations from the DEUS-FUR CDM simulation'. In: *Monthly Notices of the Royal Astronomical Society* 440.2 (Apr. 2014), pp. 1420–1434. ISSN: 0035-8711. DOI: 10.1093/mnras/stu295. arXiv: 1311.5662 (cit. on pp. 104, 109).

[159] Darren S. Reed et al. 'Toward an accurate mass function for precision cosmology'. In: *Monthly Notices of the Royal Astronomical Society* 431.2 (June 2012), pp. 1866–1882. ISSN: 0035-8711. DOI: 10.1093/mnras/stt301. arXiv: 1206.5302 (cit. on pp. 78, 109, 110, 116).

[160] Darren Reed et al. 'Evolution of the density profiles of dark matter haloes'. In: *Monthly Notices of the Royal Astronomical Society* 357.1 (Dec. 2004), pp. 82–96. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2005.08612.x. arXiv: 0312544 [astro-ph] (cit. on p. 131).

[161] Darren Reed et al. 'Evolution of the mass function of dark matter haloes'. In: *Monthly Notices of the Royal Astronomical Society* 346.2 (Dec. 2003), pp. 565–572. ISSN: 00358711. DOI: 10.1046/j.1365-2966.2003.07113.x. arXiv: 0301270 [astro-ph] (cit. on p. 134).

[162] Darren Reed et al. 'The halo mass function from the dark ages through the present day'. In: *Monthly Notices of the Royal Astronomical Society* 374.1 (July 2006), pp. 2–15. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2006.11204.x. arXiv: 0607150 [astro-ph] (cit. on pp. 123, 128, 134, 148).

[163] Alessandro B. Romeo et al. 'Discreteness Effects in ΛCDM Simulations: A Wavelet-Statistical View'. In: *The Astrophysical Journal* 686.1 (Oct. 2008), pp. 1–12. ISSN: 0004-637X. DOI: 10.1086/591236. arXiv: 0804.0294 (cit. on p. 148).

[164] Eduardo Rozo et al. 'CONSTRAINING THE SCATTER IN THE MASS-RICHNESS RELATION OF maxBCG CLUSTERS WITH WEAK LENSING AND X-RAY DATA'. In: *The Astrophysical Journal* 699.1 (July 2009), pp. 768–781. ISSN: 0004-637X. DOI: 10.1088/0004-637X/699/1/768. arXiv: 0809.2794 (cit. on p. 122).

[165] Eduardo Rozo et al. 'Cosmological Constraints from the SDSS maxBCG Cluster Catalog'. In: *The Astrophysical Journal* 708.1 (Feb. 2009), pp. 645–660. ISSN: 0004-637X. DOI: 10.1088/0004-637X/708/1/645. arXiv: 0902.3702 (cit. on p. 122).

[166] John Salmon, P. J. Quinn and M. Warren. 'Using parallel computers for very large N-body simulations: shell formation using 180 K particles.' In: *International Conference on Dynamics and Interactions of Galaxies* (1990) (cit. on p. 41).

[167] Robert J. Scherrer and Edmund Bertschinger. 'Statistics of primordial density perturbations from discrete seed masses'. In: *The Astrophysical Journal* 381 (Nov. 1991), p. 349. ISSN: 0004-637X. DOI: 10.1086/170658 (cit. on p. 144).

[168] Aurel Schneider and Romain Teyssier. 'A new method to quantify the effects of baryons on the matter power spectrum'. In: *Journal of Cosmology and Astroparticle Physics, Issue 12, article id. 049 (2015).* 12 (Oct. 2015). ISSN: 1475-7516. DOI: 10.1088/1475-7516/2015/12/049. arXiv: 1510.06034 (cit. on p. 102).

[169] Aurel Schneider et al. 'Matter power spectrum and the challenge of percent accuracy'. In: *Journal of Cosmology and Astroparticle Physics* 2016.04 (Apr. 2016), pp. 047–047. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2016/04/047. arXiv: 1503.05920 (cit. on pp. 72, 78).

[170] Aurel Schneider et al. 'Non-linear evolution of cosmological structures in warm dark matter models'. In: *Monthly Notices of the Royal Astronomical Society* 424.1 (July 2012), pp. 684–698. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2012.21252.x. arXiv: 1112.0330 (cit. on pp. 112, 148).

[171] Roman Scoccimarro. 'Transients from Initial Conditions: A Perturbative Analysis'. In: *Monthly Notices of the Royal Astronomical Society* 299.4 (Nov. 1997), pp. 1097–1118. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.1998.01845.x. arXiv: 9711187 [astro-ph] (cit. on pp. 25, 34, 109, 124).

[172] Roman Román Scoccimarro et al. 'Large-scale bias and efficient generation of initial conditions for nonlocal primordial non-Gaussianity'. In: *Physical Review D* 85.8 (Apr. 2012), p. 083002. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.85.083002. arXiv: 1108.5512 [astro-ph.CO] (cit. on p. 103).

[173] Emiliano Sefusatti et al. 'Accurate Estimators of Correlation Functions in Fourier Space'. In: *Monthly Notices of the Royal Astronomical Society* 460.4 (Dec. 2015), pp. 3624–3636. ISSN: 0035-8711. DOI: 10.1093/mnras/stw1229. arXiv: 1512.07295 (cit. on p. 120).

[174] Neelima Sehgal et al. 'The Atacama Cosmology Telescope: Cosmology from Galaxy Clusters Detected via the Sunyaev-Zel'dovich Effect'. In: *The Astrophysical Journal* 732.1 (Oct. 2010), p. 44. ISSN: 0004-637X. DOI: 10.1088/0004-637X/732/1/44. arXiv: 1010.1025 (cit. on p. 122).

[175] Uroš Seljak and Michael S. Warren. 'Large-scale bias and stochasticity of haloes and dark matter'. In: *Monthly Notices of the Royal Astronomical Society* 355.1 (Nov. 2004), pp. 129–136. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2004.08297.x (cit. on p. 105).

[176] J.˜L. Sersic. *Atlas de galaxias australes*. Cordoba, Argentina: Observatorio Astronomico, 1968, 1968 (cit. on p. 95).

[177] J.˜L. Sérsic. 'Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy'. In: *Boletin de la Asociacion Argentina de Astronomia La Plata Argentina* 6 (1963), pp. 41–+ (cit. on p. 95).

[178] Sanjib Sharma and Matthias Steinmetz. 'Multi-Dimensional Density Estimation and Phase Space Structure Of Dark Matter Halos'. In: *Monthly Notices of the Royal Astronomical Society* 373.4 (July 2005), pp. 1293–1307. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2006.11043.x. arXiv: 0507550 [astro-ph] (cit. on pp. 97, 99).

[179] Ravi K. Sheth and Giuseppe Tormen. 'Large scale bias and the peak background split'. In: *Monthly Notices of the Royal Astronomical Society* 308.1 (Jan. 1999), pp. 119–126. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.1999.02692.x. arXiv: 9901122 [astro-ph] (cit. on pp. 122, 135).

[180] M. Shimon et al. 'Constraints on the Neutrino Mass from SZ Surveys'. In: (Jan. 2012). DOI: 10.1111/j.1365-2966.2012.21563.x. arXiv: 1201.1803 (cit. on p. 122).

[181] Edwin Sirko. 'Initial conditions to cosmological N-body simulations, or how to run an ensemble of simulations'. In: *The Astrophysical Journal* 634.2 (Mar. 2005), pp. 728–743. ISSN: 0004-637X. DOI: 10.1086/497090. arXiv: 0503106 [astro-ph] (cit. on p. 147).

[182] Samuel W. Skillman et al. 'Dark Sky Simulations: Early Data Release'. In: *eprint arXiv:1407.2600* (July 2014). arXiv: 1407.2600 (cit. on pp. 74, 84, 114).

[183] R. E. Smith et al. 'Stable clustering, the halo model and non-linear cosmological power spectra'. In: *Monthly Notices of the Royal Astronomical Society* 341.4 (June 2003), pp. 1311–1332. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2003.06503.x. arXiv: 0207664 [astro-ph] (cit. on pp. 114, 140).

[184] Robert E. Smith and Laura Marian. 'What do cluster counts really tell us about the Universe?' In: *Monthly Notices of the Royal Astronomical Society* 418.2 (June 2011), pp. 729–746. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2011.19525.x. arXiv: 1106.1665 (cit. on pp. 122, 123, 148).

[185] Robert E. Smith, Ravi K. Sheth and Roman Scoccimarro. 'An analytic model for the bispectrum of galaxies in redshift space'. In: *Physical Review D* 78.2 (Dec. 2007), p. 023523. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.78.023523. arXiv: 0712.0017 (cit. on p. 140).

[186] Robert E. Smith et al. 'Precision cosmology in muddy waters: cosmological constraints and N-body codes'. In: *Monthly Notices of the Royal Astronomical Society* 440.1 (Apr. 2014), pp. 249–268. ISSN: 0035-8711. DOI: 10.1093/mnras/stu272. arXiv: 1211.6434 (cit. on pp. 116, 126, 140).

[187] Gabor Somogyi and Robert E. Smith. 'Cosmological perturbation theory for baryons and dark matter I: one-loop corrections in the RPT framework'. In: *Physical Review D* 81.2 (Oct. 2009), p. 023524. ISSN: 1550-7998. DOI: 10.1103/PhysRevD.81.023524. arXiv: 0910.5220 (cit. on p. 152).

[188] D. N. Spergel et al. 'First Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Determination of Cosmological Parameters'. In: *The Astrophysical Journal Supplement Series, Volume 148, Issue 1, pp. 175-194.* 148 (Feb. 2003), pp. 175–194. ISSN: 0067-0049. DOI: 10.1086/377226. arXiv: 0302209 [astro-ph] (cit. on p. 71).

[189] D. N. Spergel et al. 'Wilkinson Microwave Anisotropy Probe (WMAP) Three Year Results: Implications for Cosmology'. In: *The Astrophysical Journal Supplement Series* 170.2 (Mar. 2006), pp. 377–408. ISSN: 0067-0049. DOI: 10.1086/513700. arXiv: 0603449 [astro-ph] (cit. on p. 92).

[190] D. Spergel et al. 'Wide-Field InfraRed Survey Telescope-Astrophysics Focused Telescope Assets WFIRST-AFTA Final Report'. In: *eprint arXiv:1305.5422* (May 2013). arXiv: 1305.5422 (cit. on p. 71).

[191] Randall J. Splinter et al. 'Fundamental Discreteness Limitations of Cosmological N-Body Clustering Simulations'. In: *The Astrophysical Journal* 497.1 (June 1997), pp. 38–61. ISSN: 0004-637X. DOI: 10.1086/305450. arXiv: 9706099 [astro-ph] (cit. on p. 148).

[192] Volker Springel. 'The cosmological simulation code GADGET-2'. In: *Monthly Notices of the Royal Astronomical Society* 364.4 (May 2005), pp. 1105–1134. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2005.09655.x. arXiv: 0505010 [astro-ph] (cit. on pp. 74, 104, 125).

[193] Volker Springel et al. 'Simulations of the formation, evolution and clustering of galaxies and quasars'. In: *Nature* 435.7042 (June 2005), pp. 629–636. ISSN: 0028-0836. DOI: 10.1038/nature03597 (cit. on p. 104).

[194] Volker Springel et al. 'The Aquarius Project: the subhaloes of galactic haloes'. In: *Monthly Notices of the Royal Astronomical Society* 391.4 (Sept. 2008), pp. 1685–1711. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2008.14066.x. arXiv: 0809.0898 (cit. on p. 104).

[195] Joachim Stadel. 'Cosmological N–body simulations and their analysis'. PhD thesis. University of Washington, 2001 (cit. on pp. 37, 44, 59, 77, 94, 104, 125, 126, 131).

[196] Joachim Stadel et al. 'Quantifying the heart of darkness with GHALO - a multi-billion particle simulation of our galactic halo'. In: *Monthly Notices of the Royal Astronomical Society: Letters* 398.1 (Aug. 2008), pp. L21–L25. ISSN: 17453925. DOI: 10.1111/j.1745-3933.2009.00699.x. arXiv: 0808.2981 (cit. on pp. 14, 25, 45, 46, 104).

[197] R. Stanek, D. Rudd and A. E. Evrard. 'The Effect of Gas Physics on the Halo Mass Function'. In: *Monthly Notices of the Royal Astronomical Society: Letters* 394.1 (Sept. 2008), pp. L11–L15. ISSN: 17453925. DOI: 10.1111/j.1745-3933.2008.00597.x. arXiv: 0809.2805 (cit. on p. 149).

[198] Ryuichi Takahashi et al. 'REVISING THE HALOFIT MODEL FOR THE NONLINEAR MATTER POWER SPECTRUM'. In: *The Astrophysical Journal* 761.2 (Dec. 2012), p. 152. ISSN: 0004-637X. DOI: 10.1088/0004-637X/761/2/152. arXiv: 1208.2701 (cit. on pp. 113, 114).

[199] James E. Taylor and Julio F. Navarro. 'The Phase-Space Density Profiles of Cold Dark Matter Halos'. In: *The Astrophysical Journal* 563.2 (Mar. 2001), pp. 483–488. ISSN: 0004-637X. DOI: 10.1086/324031. arXiv: 0104002 [astro-ph] (cit. on pp. 92, 97, 98).

[200] Max Tegmark and Benjamin C. Bromley. 'Observational Evidence for Stochastic Biasing'. In: *The Astrophysical Journal* 518.2 (June 1999), pp. L69–L72. ISSN: 0004637X. DOI: 10.1086/312068 (cit. on p. 105).

[201] Romain Teyssier. 'Cosmological Hydrodynamics with Adaptive Mesh Refinement: a new high resolution code called RAMSES'. In: *Astronomy and Astrophysics* 385.1 (Nov. 2001), pp. 337–364. ISSN: 0004-6361. DOI: 10.1051/0004-6361:20011817. arXiv: 0111367 [astro-ph] (cit. on pp. 74, 104, 116).

[202] Romain Teyssier et al. 'Full-Sky Weak Lensing Simulation with 70 Billion Particles'. In: *Astronomy and Astrophysics* 497.2 (July 2008), pp. 335–341. ISSN: 0004-6361. DOI: 10.1051/0004-6361/200810657. arXiv: 0807.3651 (cit. on p. 104).

[203] Jeremy L Tinker et al. 'Toward a halo mass function for precision cosmology: the limits of universality'. In: *The Astrophysical Journal* 688.2 (Mar. 2008), pp. 709–728. ISSN: 0004-637X. DOI: 10.1086/591439. arXiv: 0803.2706 (cit. on pp. 123, 128, 131, 148).

[204] Erik J. Tollerud et al. 'Hundreds of Milky Way Satellites? Luminosity Bias in the Satellite Luminosity Function'. In: *The Astrophysical Journal* 688.1 (June 2008), pp. 277–289. ISSN: 0004-637X. DOI: 10.1086/592102. arXiv: 0806.4381 (cit. on p. 92).

[205] Michele Trenti et al. 'How well do cosmological simulations reproduce individual-halo properties?' In: *The Astrophysical Journal* 711.2 (Jan. 2010), pp. 1198–1207. ISSN: 0004-637X. DOI: 10.1088/0004-637X/711/2/1198. arXiv: 1001.5037 (cit. on pp. 128, 146).

[206] K. Vanderlinde et al. 'Galaxy Clusters Selected with the Sunyaev-Zel'dovich Effect from 2008 South Pole Telescope Observations'. In: *The Astrophysical Journal* 722.2 (Mar. 2010), pp. 1180–1196. ISSN: 0004-637X. DOI: 10.1088/0004-637X/722/2/1180. arXiv: 1003.0003 (cit. on p. 122).

[207] A. Vikhlinin et al. 'Chandra Cluster Cosmology Project III: Cosmological Parameter Constraints'. In: *The Astrophysical Journal* 692.2 (Dec. 2008), pp. 1060–1074. ISSN: 0004-637X. DOI: 10.1088/0004-637X/692/2/1060. arXiv: 0812.2720 (cit. on p. 122).

[208] Jie Wang and Simon D. M. White. 'Discreteness effects in simulations of Hot/Warm dark matter'. In: *Monthly Notices of the Royal Astronomical Society* 380.1 (Feb. 2007), pp. 93–103. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2007.12053.x. arXiv: 0702575 [astro-ph] (cit. on p. 148).

[209] Limin Wang and Paul J. Steinhardt. 'Cluster Abundance Constraints on Quintessence Models'. In: *The Astrophysical Journal* 508.2 (Apr. 1998), pp. 483–490. ISSN: 0004-637X. DOI: 10.1086/306436. arXiv: 9804015 [astro-ph] (cit. on p. 122).

[210] Sheng Wang et al. 'Weighing Neutrinos with Galaxy Cluster Surveys'. In: *Physical Review Letters* 95.1 (May 2005), p. 011302. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.95.011302. arXiv: 0505390 [astro-ph] (cit. on p. 122).

[211] M. S. Warren and J. K. Salmon. 'A parallel hashed Oct-Tree N-body algorithm'. In: *Proceedings of the 1993 ACM/IEEE conference on Supercomputing - Supercomputing '93*. New York, New York, USA: ACM Press, 1993, pp. 12–21. ISBN: 0818643404. DOI: 10.1145/169627.169640 (cit. on p. 73).

[212] Michael S. Warren and Michael S. '2HOT: An Improved Parallel Hashed Oct-Tree N-Body Algorithm for Cosmological Simulation'. In: *eprint arXiv:1310.4502* (Oct. 2013). DOI: 10.1145/2503210.2503220. arXiv: 1310.4502 (cit. on pp. 74, 75).

[213] Michael S. Warren et al. 'Precision Determination of the Mass Function of Dark Matter Halos'. In: *The Astrophysical Journal* 646.2 (June 2005), pp. 881–885. ISSN: 0004-637X. DOI: 10.1086/504962. arXiv: 0506395 [astro-ph] (cit. on pp. 123, 128, 129).

[214] William A. Watson et al. 'The halo mass function through the cosmic ages'. In: *Monthly Notices of the Royal Astronomical Society* 433.2 (Dec. 2012), pp. 1230–1245. ISSN: 0035-8711. DOI: 10.1093/mnras/stt791. arXiv: 1212.0095 (cit. on pp. 123, 148).

[215] Peter Watts and Andrew Taylor. 'Evolution of the probability distribution function of galaxies in redshift-space'. In: *Monthly Notices of the Royal Astronomical Society* 320.1 (June 2000), pp. 139–152. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2001.03948.x. arXiv: 0006192 [astro-ph] (cit. on p. 144).

[216] Hao-Yi Wu, Andrew R. Zentner and Risa H. Wechsler. 'The Impact of Theoretical Uncertainties in the Halo Mass Function and Halo Bias on Precision Cosmology'. In: *The Astrophysical Journal* 713.2 (Oct. 2009), pp. 856–864. ISSN: 0004-637X. DOI: 10.1088/0004-637X/713/2/856. arXiv: 0910.3668 (cit. on p. 122).

[217] Ya. B. Zel'dovich. 'Gravitational instability: An approximate theory for large density perturbations.' In: *Astronomy and Astrophysics* 5 (1970) (cit. on pp. 25, 124).

[218] Marcel Zemp et al. 'An optimum time-stepping scheme for N-body simulations'. In: *Monthly Notices of the Royal Astronomical Society* 376.1 (June 2006), pp. 273–286. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2007.11427.x. arXiv: 0606589 [astro-ph] (cit. on p. 94).

[219] HongSheng Zhao. 'Analytical Models For Galactic Nuclei'. In: *Monthly Notices of the Royal Astronomical Society* 278.2 (Sept. 1995), pp. 488–496. ISSN: 0035-8711. DOI: 10.1093/mnras/278.2.488. arXiv: 9509122 [astro-ph] (cit. on p. 95).

[220] F. Zwicky. 'Die Rotverschiebung von extragalaktischen Nebeln'. In: *Helvetica Physica Acta* 6 (1933), pp. 110–127 (cit. on p. 11).